Miguel Filipe Alves da Silva João

Licenciado em Engenharia Electrotécnica e de Computadores

Controlo Neuro Difuso com Consequentes no Espaço de Estados

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Orientador: Paulo José Carrilho de Sousa Gil, Professor Auxiliar,
Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa

Júri

Presidente:	Doutor André Teixeira Bento Damas Mora - FCT/UNL
Arguente:	Doutor Luís Filipe Figueira de Brito Palma - FCT/UNL
Vogal:	Doutor Paulo José Carrilho de Sousa Gil - FCT/UNL

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2019

Controlo Neuro Difuso com Consequentes no Espaço de Estados

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Agradecimentos

Agradeço à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, pela oportunidade e por se mostrarem disponíveis durante todo o meu percurso académico.

Agradeço ao professor Paulo Gil, pela sua paciência e disponibilidade na orientação desta dissertação e por todos os desafios propostos ao longo deste trabalho.

Agradeço também ao aluno de doutoramento Fábio Januário por todo o auxílio prestado em fases cruciais do desenvolvimento do trabalho e pela sua amizade.

Queria agradecer a todos os meus colegas que me ajudaram a concluir os meus estudos, com especial apreço Inês Martins, Diogo Alves, David Crispim e Frederico Anastácio sendo essenciais quer na conclusão do meu percurso académico nesta instituição, quer na minha vida. Agradeço a todos os meus colegas e amigos que não estando aqui discriminados sempre me apoiaram nos melhores e piores momentos.

Agradeço à minha família por todo o apoio e carinho com especial agradecimento aos meus pais Isabel João e João Silva, por terem me facultado a hipótese de estudar, e por me terem ajudado a concluir mais uma etapa da minha vida.

Resumo

A presente dissertação tem como principal objectivo a implementação de um protótipo de controlo que visa solucionar o problema de controlo de seguimento, utilizando uma abordagem neuro difusa baseada no modelo *Adaptive Neuro Fuzzy System*-ANFIS e em *Linear Matrix Inequality*-LMI. Com este fim desenvolveu-se o protótipo que utiliza a aproximação por modelos locais lineares recorrendo à inferência difusa de Takagi Sugeno representada por uma rede neuronal recursiva. Assim utiliza-se as LMI's sob o conceito *Parallel Distributed Compensation*-PDC para calcular as matrizes de retroacção associadas a cada modelo local. De forma a poder validar o protótipo desenvolvido nesta dissertação utilizou-se o sistema *Multiple Input Multiple Output*-MIMO Amira DTS200, sendo este constituído por um sistema de 3 tanques. A solução mostrou-se válida para este processo, apresentando um erro em regime permanente nulo, e um RMSE inferior a 5 % e um MSI inferior a 0,1 %. O protótipo de controlo foi ainda comparado com um controlador PID difuso com inferência do tipo Mandami, tendo-se revelado superior no que diz respeito à análise de erros. O Controlador desenvolvido nesta dissertação permitiu assim solucionar o problema de controlo de seguimento, de uma forma considerada robusta.

Palavras-chave: Controlo Neuro Difuso; ANFIS; LMI's; Inferência T-S; Redes neuronais; PDC; RSME; Modelos difusos Takagi Sugeno;

Abstract

The main objective of the present dissertation consists in the implementation of a control prototype to solve the tracking control problem using a neuro fuzzy approach based on the Adaptive Neuro Fuzzy System-ANFIS and Linear Matrix Inequality-LMI's model. For this purpose, a prototype was developed using the linear local models approximation by linear local models using the fuzzy Takagi Sugeno inference represented by a recursive neuronal network. The linear matrix inequality under the concept of Parallel Distributed Compensation-PDC is used to calculate the feedback matrix gains associated to each local state space model. In order to validate the prototype developed in this dissertation the Multiple Input Multiple Output -MIMO, Amira DTS200 was used. The system Amira DTS200 consists of a system of 3 tanks. The solution proved to be valid for this process, having a zero steady state error, and an RMSE of less than 5% and an MSI of less than 0.1%. The control prototype was further compared to a Mandami-type PID fuzzy controller and it showed to be superior for error analysis. The controller developed in this dissertation solved the tracking control problem.

Keywords: LMI's; PDC; Root Mean Square error; Fuzzy Takagi Sugeno Models; Neural Network; Neuro Fuzzy Controller; ANFIS;

Índice

Lista de Figuras

Lista de Tabelas

Siglas

AEN	*Action State Evaluation Network.*
ANFIS	*Adaptive Network based Fuzzy Inference System.*
ASN	*Action Selection Network.*
EFuNN/dmEFuNN	*Dynamic/ Evolving Fuzzy Neural Network.*
FALCON	*Fuzzy Adaptive Learning Control Network.*
FINEST	*Fuzzy Inference and Neural Network in Fuzzy Inference Software.*
FUN	*Fuzzy Net.*
GARIC	*Generalized Approximate Reasoning Based Intelligence Control.*
HDS	*Hybrid Dynamical Systems.*
LMI's	*Linear Matrix Inequality.*
LSE	*Least Square Estimation.*
MSE	*Mean Square Error.*
MSI	*Mean Square of Control Action Increment.*
NEFCLASS	*Neuro Fuzzy Classification.*
NEFCON	*Neuronal Fuzzy Controller.*
NEFPROX	*Neuro Fuzzy Function Approximation.*
NFN	*Fuzzy Neural Network.*
PDC	*Parallel Distributed Compensation.*
PSO	*Particle Swarm Optimization.*
RMSE	*Root Mean Square Error.*

SONFIN *Self Constructing Neural Fuzzy Inference Network*.

T-S *Modelos Takagi e Sugeno*.

Capítulo 1

Introdução

Na indústria hoje em dia e cada vez mais se caminha para uma automatização, e por sua vez para sistemas distribuídos, integrando inteligência artificial. Tal leva a que o controlo cada vez mais evolua para uma maior utilização de controladores neuro difusos Keller et al. 2014. O grande interesse no uso deste tipo de controladores deve-se ao facto da facilidade de os integrar e de os parametrizar de forma que estes se adaptem a comutações de sistema. Tal possibilita que a indústria possa ter sistemas mais distribuídos e com várias funções sem ter necessidade de parametrizar ou alterar o controlador.

1.1 Motivação

Cada vez mais com a evolução tecnológica, a indústria pretende caminhar para sistemas distribuídos, com alguma inteligência artificial. Assim sendo, o controlo que faz a ligação do software de alto nível ao *hardware* necessita de se adaptar a este novo paradigma. Com este fim desenvolveram-se arquitecturas inspiradas no cérebro humano e a forma como ele comunica e reage a diferentes estímulos, tendo vindo a desenvolver-se as redes neuronais e a lógica difusa que tem como objectivo emular em certa parte o comportamento do cérebro humano, permitindo ter uma indústria mais distribuída com uma certa inteligência artificial. Estas arquitecturas permitem uma maior facilidade no controlo de sistemas não lineares. Se forem utilizadas as duas metodologias em conjunto poderá ter-se então um controlo neuro difuso, tendo a parte neuronal a armazenar o conhecimento e a parte difusa com o poder de decisão. Assim sendo, a rede neuronal tem como objectivo influenciar as regras difusas de forma que o controlador vá alterando o seu poder de decisão à medida que a rede neuronal vai adquirindo conhecimento. A principal motivação para a realização desta dissertação prende-se com as potencialidades deste tipo de algoritmos

que utilizam a combinação das redes neuronais com a lógica difusa, e que permitem uma maior descentralização da industria, possibilitando uma maior adaptação e aumentos de eficiência.

1.2 Objetivos

Este trabalho consiste na implementação de controladores neuro difusos com o objectivo de resolver o problema de controlo de seguimento, para sistemas não lineares. Para este fim será utilizado o sistema dos três tanques com diversas configurações de forma a validar a arquitectura proposta para a solução do problema.

Os principais objectivos são:

1. Revisão Bibliográfica sobre modelação com modelos neuro difusos e projeto de controladores.
2. Desenvolvimento de um protótipo de controlador neuro difuso para resolução do problema de controlo de seguimento para sistemas não lineares.
3. Validação do protótipo desenvolvido por recurso à utilização do sistema MIMO Amira DTS200 sendo este constituído por um sistema de três tanques.
4. Cálculo de métricas de erro para avaliar o desempenho do controlador.
5. Comparação do protótipo de controlo desenvolvido com um controlador PID difuso com inferência do tipo Mandani.

1.3 Contribuições

Nesta dissertação encontram-se algumas contribuições para a comunidade cientifica, tais como a criação de um modelo neuro difuso com inferência difuso do tipo Takagi e Sugeno e com consequentes em espaço de estados. Esta estrutura tem como base a estrutura já desenvolvida ANFIS. Este modelo descreve assim o sistema Amira DTS 200, um sistema composto por três tanques. Foi ainda idealizado um algoritmo de treino para este modelo utilizando um algoritmo PSO para otimizar os antecedentes e um algoritmo PSO com backpropagation para otimizar os consequentes do modelo neuro difuso proposto. Desenvolveu-se também um controlador neuro difuso baseado também na arquitectura de controlo ANFIS com a adição de uma camada a rede neuronal e alteração dos consequentes para espaço de estados. Por fim desenvolveu-se um sistema anti-windup de forma a que acção de controlo integral não satura-se.

1.4 Organização

Esta dissertação encontra-se dividida em 5 capítulos, incluindo o presente. No capítulo 2 são introduzidos os conceitos base da modelação difusa e por sua vez a modelação

neuro difusa utilizando a estrutura difusa de Takagi Sugeno. Neste capítulo são ainda apresentadas várias arquitecturas de modelos neuro difusos que permitem combinações diferentes entre as redes neuronais e os sistemas difusos.

No capítulo 3 é abordado o projecto de controladores neuro difusos bem como algumas definições de estabilidade de sistemas dinâmicos. Apresenta-se também o método das *Linear Matrix Inequality* (LMI's) de forma a garantir a estabilidade em anel aberto e fechado, bem como a arquitectura neuro difusa a utilizar para a construção do controlador a implementar neste trabalho.

No capítulo 4 é descrito o processo Amira DTS200 de forma a poder testar e validar a arquitectura proposta neste trabalho. Procede-se à construção de um modelo não linear através de modelos lineares e à implementação do protótipo neuro difuso com base na arquitectura ANFIS. Constrói-se ainda um sistema de *anti-windup* de forma a melhorar o desempenho do controlador. Apresentam-se também todos os testes e validações para a construção do modelo não linear bem como a construção do controlador neuro difuso. Serão também comparados e analisados os resultados destas simulações. Para além disso, será ainda efectuada a comparação do protótipo de controlo a desenvolver com um controlador PID Difuso com inferência do tipo Mandani.

No capítulo 5 são apresentadas as principais conclusões do trabalho realizado, e ainda algumas perspectivas de trabalho futuro relativas à extensão do presente trabalho.

Capítulo 2

Modelação com modelos neuro-difusos

2.1 Introdução

Desde o inicio do seu desenvolvimento em 1965 Zadeh 1965 os modelos difusos têm vindo a ser usados nos mais variados domínios científicos incluindo, na medicina Teodorescu et al. 1999, nos processos químicos e industriais Emami 2010, telecomunicações Ghosh et al. 1998, robótica Scharf et al. 1987; Xiong et al. 2008, entre outros. Uma área que beneficiou especialmente foi a do controlo uma vez que grande parte dos sistemas reais e unidades industriais possuem não linearidades, parâmetros que variam com o tempo, erros de modelação e serem sujeitos a perturbações sobre as quais não existe muito conhecimento.

O controlo difuso baseado em *Modelos Takagi e Sugeno* (T-S) é uma abordagem muito adequada devido ao aumento crescente da complexidade dos sistemas reais e dos critérios de projecto industrial cada vez mais restritivos e exigentes. Os sistemas difusos assumiram um papel de relevo na indústria ao longo dos anos Hirota e Sugeno 1995 mas desenvolver sistemas difusos de adequado desempenho não é tarefa fácil, além da dificuldade que muitas vezes existe em encontrar funções de pertença e regras apropriadas. A utilização de redes neuronais, com eficientes algoritmos de aprendizagem, surgiu assim adequada como suporte a dar aos sistemas difusos.

Os primeiros estudos efectuados com modelos neuro-difusos datam dos anos 90 do século passado, Jang 1993; Lin e Lee 1991a; Berenji e Khedkar 1992b, com trabalhos de Jang, Lin e Lee ou ainda Berengi e Khedkar a titulo de exemplos. As redes neuronais surgem como forma de aplicar algoritmos de aprendizagem aos sistemas difusos havendo distintas formas de implementação de sistemas neuro difusos.

Neste capítulo aborda-se a modelação difusa de Takagy-Sugeno e a forma de construção do modelo difuso. Abordam-se ainda os modelos neuro-difusos bem como as suas principais formas de implementação.

2.2 Modelação difusa Takagi-Sugeno

A modelação com base numa estrutura difusa Takagi-Sugeno (modelos difusos tipo T-S), tem como base de regras, *IF-Then*, sendo que cada regra corresponde a um regime de funcionamento linear de um sistema não linear. Estas regras que representam relações locais de entrada-saída foram propostas por Takagi e Sugeno tal como descrito em Takagi e Sugeno 1985; Tanaka e Wang 2001. Assim sendo a modelação difusa consiste na junção de vários modelos lineares em diferentes pontos de operação.

Os sistemas difusos podem ser contínuos ou discretos e a forma das suas regras é do seguinte tipo, 2.1, 2.2:

Sistemas fuzzy contínuos:

$$\begin{aligned} &\textbf{IF } z_1(t) \textit{ is } M_{i1} \textit{ and } ... \textit{ and } z_p(t)) \textit{ is } M_{ip}, \\ &\textbf{THEN} \begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i\boldsymbol{x}(t) + \boldsymbol{B}_i\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}_i\boldsymbol{x}(t) \end{cases} \quad i = 1, 2, ..., r. \end{aligned} \tag{2.1}$$

Sistemas fuzzy discretos:

$$\begin{aligned} &\textbf{IF } z_1(k) \textit{ is } M_{i1} \textit{ and } ... \textit{ and } z_p(k)) \textit{ is } M_{ip}, \\ &\textbf{THEN} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}_i\boldsymbol{x}(k) + \boldsymbol{B}_i\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \boldsymbol{C}_i\boldsymbol{x}(k) \end{cases} \quad i = 1, 2, ..., r. \end{aligned} \tag{2.2}$$

onde $\boldsymbol{u}(\mathrm{t}) \in \mathbf{R^m}$ e $\boldsymbol{y}(\mathrm{t}) \in \mathbf{R^q}$ correspondem respectivamente ao vector de entrada e ao vector de saída, e por sua vez $\boldsymbol{x}(\mathrm{t}) \in \mathbf{R^n}$ corresponde ao vector de estado. As matrizes no espaço de estados, $\mathbf{A_i} \in \mathbf{R^{n\times n}}$, $\mathbf{B_i} \in \mathbf{R^{n\times m}}$ e $\mathbf{C_i} \in \mathbf{R^{q\times n}}$, são variáveis conhecidas que podem ser representadas como funções ou simplesmente como um único valor. Pode-se ainda constatar que $\mathbf{M_{ij}}$ corresponde aos conjuntos difusos, ou seja a um par (u,m), sendo u um conjunto e m função de pertença, e r corresponde ao número de regras implementadas para o modelo.

Assim sendo a saída do modelo difuso, para um par $(\boldsymbol{u}(\mathrm{t}),\boldsymbol{x}(\mathrm{t}))$, toma a seguinte forma 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9:

Sistemas fuzzy contínuos:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \frac{\sum_{i=1}^{r} w_i(z(t)) \{\mathbf{A_i x}(t) + \mathbf{B_i u}(t)\}}{\sum_{i=1}^{r} w_i(z(t)} \\ &= \sum_{i=1}^{r} h_i(z(t)) \{\mathbf{A_i x}(t) + \mathbf{B_i u}(t)\} \end{aligned} \tag{2.3}$$

$$\begin{aligned} \boldsymbol{y}(t) &= \frac{\sum_{i=1}^{r} w_i(z(t)) \mathbf{C_i x}(t)}{\sum_{i=1}^{r} w_i(z(t)} \\ &= \sum_{i=1}^{r} h_i(z(t)) \mathbf{C_i x}(t) \end{aligned} \tag{2.4}$$

Sistemas fuzzy discretos:

$$\begin{aligned} \boldsymbol{x}(t+1) &= \frac{\sum_{i=1}^{r} w_i(z(t))\{\mathbf{A_i}\mathbf{x}(t)+\mathbf{B_i}\mathbf{u}(t)\}}{\sum_{i=1}^{r} w_i(z(t)} \\ &= \sum_{i=1}^{r} h_i(z(t))\{\mathbf{A_i}\mathbf{x}(t)+\mathbf{B_i}\mathbf{u}(t)\} \end{aligned} \tag{2.5}$$

$$\begin{aligned} \boldsymbol{y}(t) &= \frac{\sum_{i=1}^{r} w_i(z(t))\mathbf{C_i}\mathbf{x}(t)}{\sum_{i=1}^{r} w_i(z(t)} \\ &= \sum_{i=1}^{r} h_i(z(t))\mathbf{C_i}\mathbf{x}(t) \end{aligned} \tag{2.6}$$

Sendo que w_i, $z(t)$, h_i é dado por:

$$\boldsymbol{z}(t) = [z_1(t) \quad z_2(t) \quad ... \quad z_p(t)] \tag{2.7}$$

$$\mathbf{w_i}(z(t)) = \prod_{j=1}^{p} M_{ij}(z_j(t)) \tag{2.8}$$

$$\mathbf{h_i}(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^{r} w_i(z(t)} \tag{2.9}$$

2.3 Construção do modelo difuso

Considerando as equações que definem os modelos fuzzy, vai-se abordar a forma de construir este modelo esquematizado na figura 2.1:

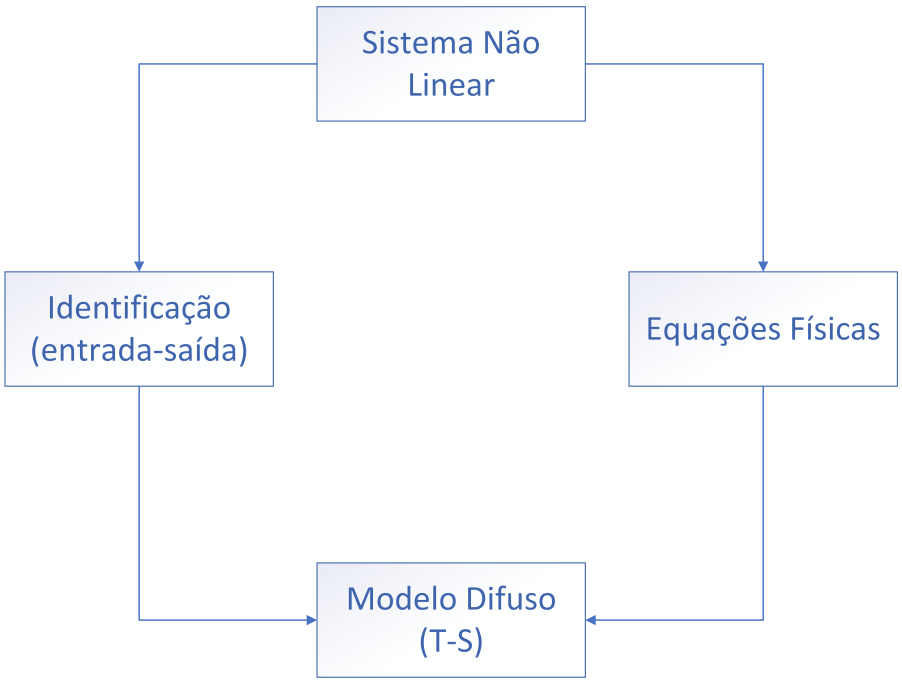

Figura 2.1: Diagrama da Modelação Difusa.

Assim sendo para a construção de um modelo difuso existem duas abordagens:

1. Identificação através do conjunto de entrada e saída, sendo dividido em conjunto de treino e validação, de forma a construir o modelo.

2. Através das equações físicas que definem o sistema em questão.

Quando se pretende construir um modelo através da abordagem 1 utilizando os dados de entrada e saída recorre-se a dois aspectos fundamentais, a identificação da estrutura, e a identificação dos parâmetros.

Para mais informação sobre os procedimentos de identificação sugere-se a leitura do trabalho de Sugeno e Kang Sugeno e Kang 1988. Esta abordagem é adequada para problemas que são difíceis de representar por modelos analíticos e/ou físicos.

Em relação à abordagem 2 esta consiste em utilizar algoritmos matemáticos como por exemplo o método de Lagrange, ou físicos como é o caso da lei da conservação da massa, de forma a modelar o sistema, tendo assim uma ou mais equações que o descrevem.

De forma a operacionalizar o modelo T-S é necessário definir o número de entradas do processo, a escolha da estrutura do modelo que envolve a determinação da quantidade de regras, o tipo de funções de pertença e tipo de consequente e estimação dos parâmetros das funções de pertença e dos consequentes, referentes aos parâmetros para cada regra.

A selecção de entradas do processo é uma questão pertinente e abordada por diversos autores Sáez e Zúñiga 2005; Hadjili e Wertz 2002.

No que diz respeito à estrutura, esta tem um papel preponderante na construção e na eficiência do modelo T-S. Para a escolha da estrutura, Du e Zhang Du e Zhang 2008 utilizam uma abordagem via algoritmos genéticos de modo a obter uma estrutura do modelo T-S. Os autores mostram que o modelo T-S desenvolvido pode identificar de forma satisfatória o sistema não linear com um número aceitável de regras e de sinais de entrada (*inputs*) adequados, através do ajuste simultâneo das entradas, da estrutura de regras e dos parâmetros das funções de pertença.

Uma abordagem sistemática que permite a representação exacta, local ou global (se bem que por vezes seja difícil determinar sectores globais em sistemas não lineares) foi desenvolvida pela primeira vez em Kawamoto et al. 1992 e é conhecida como abordagem de não linearidade sectorial (*sector nonlinearity*). Uma outra abordagem que leva a um número reduzido de regras para obter modelos fuzzy T-S é a chamada aproximação local em espaços de partições difusas. Neste caso, o modelo derivado é uma aproximação do sistema não linear. A obtenção dos modelos lineares pode ser feita a partir da escolha adequada de pontos de linearização para o sistema não linear.

2.4 Modelos neuro-difusos

Nesta secção aborda-se os modelos difusos T-S e a sua evolução para modelos neuro-difusos, e respectiva inferência difusa. Contudo, antes de descrever a estrutura de um modelo neuro-difuso, é importante definir alguns conceitos em relação às redes neuronais Kriesel 2005:

- **Sinapse:** Representa a ligação entre neurónios tendo um peso associado a cada ligação, designado por $w_{r,i} \in \mathbb{R}$.
- **Neurónio:** É o elemento principal numa rede neuronal. Todos os neurónios respondem a um estimulo produzindo uma saída O_r, e por sua vez todos os neurónios recebem como entrada, a saída da camada anterior O^i_{r-1} que estão conectados pelas sinapses. Os neurónios têm como função agrupar toda a informação recebida da camada anterior, tal como representado na equação 2.10, e por sua vez utilizar uma função de activação de forma a construir a saída a ser enviada para a camada seguinte, de acordo com a equação 2.11.

$$net_{r,j} = fprop(O_{r-1}, W_{r,j}) = \sum_{i=1}^{n} (O^i_{r-1} \times w^i_{r,j}) \tag{2.10}$$

$$O_r = f_{act}(net_{r,j}, O_{r-1}, \theta_j) \tag{2.11}$$

Analisando as equações 2.10 e 2.11, o índice i corresponde a uma camada anterior e por sua vez o índice j corresponde à camada actual. Em relação às funções de activação

pode-se ter vários tipos de funções como por exemplo, bipolar, linear, sigmóide, tangente hiperbólica, entre outras Miguel e Oliveira 2013; Kriesel 2005. Para a escolha da função de activação é sempre preciso que a função seja diferenciavel de forma a se poder implementar um algoritmo de retropropagação Goh 1995; Hunt et al. 1992; HECHT-NIELSEN 1992.

Existem várias arquitecturas de modelos neuro-difusos, que permitem combinações diferentes entre as redes neuronais e os sistemas difusos. Em geral estas combinações podem ser divididas da seguinte forma Nauck et al. 1997 :

- **Sistema neuro-difuso Cooperativo:** Nestes sistemas, as redes neuronais funcionam como um pré-processamento, sendo sub-blocos do sistema difuso. Têm como objectivo determinar os conjuntos difusos e/ou as regras Kasabov 1998; Czogala e Leski 2000. Após o cálculo destes sub conjuntos do sistema difuso serem determinados, a rede neuronal é ignorada, ficando exclusivamente um sistema difuso.

- **Sistema neuro-difuso Concorrente:** Neste tipo de modelos a rede neuronal e o sistema difuso trabalham cooperativamente na mesma tarefa, ou seja a rede neuronal ou pré-processa as entradas ou pós processa as saídas do sistema difuso.

- **Sistema neuro-difuso Híbrido:** Neste tipo de sistemas a rede neuronal tem como objectivo determinar algumas componentes do sistema difuso iterativamente, como por exemplo, a base de regras, os conjuntos difusos, os pesos associados a cada regra.

Devido ao facto da abordagem neuro-difusa híbrida apresentar alguns benefícios em relação às anteriores, esta tem vindo a ser mais utilizada, razão pela qual irá ser dado um maior relevo a esta abordagem de conjugação de redes neuronais com modelos difusos. Um sistema neuro-difuso híbrido é um sistema difuso que usa algoritmos de aprendizagem para determinar os conjuntos difusos e as regras difusas, através de processamento de dados de treino. A utilização das redes neuronais neste sistema permite visualizar o fluxo de dados através do sistema e os sinais de erro que são utilizados para actualizar os conjuntos difusos e as regras difusas. Estas permitem ainda que diferentes modelos sejam comparados e que as diferenças estruturais sejam facilmente percebidas, o que faz com que as redes neuronais em conjunto com modelos difusos possam ser aplicadas a sistemas comutados.

Na literatura encontram-se diversas arquitecturas neuro-difusas híbridas, sendo que as mais importantes são:

- ***Fuzzy Adaptive Learning Control Network* (FALCON):** Este tipo de arquitectura faz uso de um algoritmo de aprendizagem híbrido constituído por uma aprendizagem não supervisionada que tem por objectivo a definição da base de regras iniciais e funções de pertença e o algoritmo de aprendizagem para optimização e ajuste dos parâmetros das funções de pertença. O sistema tem um total de cinco camadas. Os neurónios da primeira camada, que são os de entrada, representam as variáveis

linguísticas. Na camada de saída, camada 5, existem dois tipos de neurónios, os que permitem que o algoritmo aprenda e os que servem para actuar o sistema. Os neurónios das camadas 2 e 4 funcionam como funções de pertença para representar a variável linguística respectiva. Cada neurónio da camada 3 representa uma regra difusa. A camada 2 pode ser representada por um único neurónio ou por um conjunto de neurónios multi-camada razão pela qual o número total de camadas nesta arquitectura possa ser constituída por um número superior a 5 níveis Lin e Lee 1991b.

- ***Generalized Approximate Reasoning Based Intelligence Control* (GARIC):** O sistema neuro-difuso implementado pela arquitectura GARIC utiliza duas redes neuronais, a *Action Selection Network* (ASN) e a *Action State Evaluation Network* (AEN). O ASN é constituído por cinco camadas sem ponderação entre elas. A primeira camada tem como objectivo guardar os valores linguísticos das variáveis de entrada e a segunda camada representa os nós das regras difusas, usando o operador *softmin* para calcular o grau de compatibilidade de cada regra. A terceira camada implementa a união dos antecedentes numa única regra e um neurónio na camada 3 corresponde a uma regra pertencente à base de regras. A camada 4 tem como objectivo receber as regras que dispararam de forma a desdifuzificar categorizando os consequentes. A camada cinco é constituída por um número de neurónios igual ao número de saídas do sistema e tem como objectivo calcular valores numéricos para actuar no sistema. O AEN é um avaliador adaptativo das acções do ASN. Para este fim esta rede neuronal utiliza um algoritmo de recompensa/punição para actualização dos pesos desta rede permitindo comparar o resultado entre a rede ASN e esta Berenji e Khedkar 1992a.

- ***Neuronal Fuzzy Controller* (NEFCON):** A arquitectura NEFCON foi concebida para implementar um sistema de inferência difusa do tipo Mandami. Nesta arquitectura as ligações são ponderadas com conjuntos difusos e regras utilizando os mesmos antecedentes, que são pesos partilhados e asseguram a integridade da base de regras. A arquitectura é constituída por 3 camadas sendo que a camada de entrada assume a função de difuzificação, a camada de saída é responsável pela desdifuzificação e a interface lógica representada pela função de propagação. O processo de aprendizagem desta arquitectura baseia-se no algoritmo de retropropagação. Esta arquitectura tem duas variantes o *Neuro Fuzzy Classification* (NEFCLASS) para tarefas de classificação e o *Neuro Fuzzy Function Approximation* (NEFPROX) para tarefas de aproximação de funções Nauck et al. 1997.

- ***Fuzzy Inference and Neural Network in Fuzzy Inference Software* (FINEST):** A arquitectura FINEST tem 3 aspectos fundamentais que são o *modus ponens* melhorado, um mecanismo que permite a sintonização de predicativos difusos e um software de correcção de erros e sintonização. Tano, Oyama e Arnould Tano et al. 1996 fornecem

uma descrição do *software* e descrevem a combinação da inferência difusa com as redes neuronais o que possibilita que o FINEST auto sintonize o próprio método de inferência difusa.

- ***Fuzzy Net* (FUN):** A rede é constituída por uma camada de entrada, uma camada de saída e três camadas escondidas. Os neurónios de cada camada têm diferentes funções de activação, representando diferentes níveis no cálculo da inferência difusa. As funções de activação podem ser escolhidas de forma individual para diferentes problemas. A rede é inicializada com uma base de regras difusas e as suas correspondentes funções de pertença. As variáveis de entrada do processo de trabalho são armazenadas na camada de entrada. A primeira camada escondida contém as funções de pertença e efectua a difuzificação das variáveis de entrada. Na segunda camada escondida a função de activação dos neurónios corresponde à função AND, e nesta camada calcula-se a conjugação (fuzzy-AND). Nos neurónios da terceira camada escondida encontram-se as funções de pertença das variáveis de saída e a função de activação é a função (fuzzy-OR) correspondendo à desfuzificação das variáveis. De seguida estas são armazenadas na camada de saídaSulzberger et al. 1993.

- ***Fuzzy Neural Network* (NFN):** Esta arquitectura utiliza neurónios lógicos que implementam funções do tipo norma-T e conorma-T. As variáveis de entrada são difuzificadas e são posteriormente ponderadas e ligadas a neurónios do tipo AND que são colocados para que todas as combinações de variáveis de entrada sejam formadas. Cada neurónio do tipo AND é ligado a um e só um neurónio do tipo OR, neurónios esses que são colocados para todas as classes de saída. As ponderações das ligações dos termos difusos aos neurónios AND são inicializadas com zero e as ponderações das ligações dos neurónios AND aos OR tomam o valor unitário na inicialização. Estas ponderações correspondem a graus de incerteza das regras. Por sua vez a aprendizagem é feita por um algoritmo de duas fases. Uma primeira fase de auto-organização para adaptação dos pesos que ligam os termos difusos aos neurónios AND (aprendizagem das funções de pertença dos antecedentes de cada regra) e uma segunda fase que utiliza um esquema de supervisão para a adaptação dos consequentes das regras através da actualização dos pesos que ligam os neurónios AND aos OR Figueiredo e Gomide 1999.

- ***Dynamic/ Evolving Fuzzy Neural Network* (EFuNN/dmEFuNN):** Nesta arquitectura a primeira camada implementa a desdifuzificação das variáveis de entrada, armazenando estas na segunda camada para se calcular os graus das funções de pertença. A terceira camada representa a base de regras, sendo cada nó de regras definido por dois vectores de ponderações de ligação ajustados por um algoritmo de aprendizagem híbrida. As saídas das funções de pertença são combinadas com os dados de entrada com um determinado grau calculado na quarta camada. A quinta

camada é responsável pela desfuzificação e determina o valor numérico para as variáveis de saída. Foi também criada uma versão modificada desta arquitectura denominada de dmEFuNN que estima regras difusas do tipo Takagi-Sugeno baseadas no algoritmo dos mínimos quadráticos enquanto que a arquitectura EFuNN implementa regras difusas do tipo de Mandani. Kasabov e Song 1999.

- ***Self Constructing Neural Fuzzy Inference Network* (SONFIN):** Nesta arquitectura a estrutura de identificação de antecedentes, o espaço de entradas é flexível de acordo com um algoritmo baseado no alinhamento de classes. Já em relação a identificação dos consequentes, só um valor será seleccionado por um algoritmo de classificação. Para a optimização das estruturas de antecedentes e consequentes, são ajustados pelo algoritmo de retropropagação e pelo método da média dos mínimos quadráticos respectivamente. Esta arquitectura tem como objectivo implementar um sistema difuso de inferência do tipo Takagi-Sugeno modificado Juang e Lin 1998.

- *Adaptive Network Based Fuzzy Inference System* (**ANFIS**)**:** Este tipo de arquitectura tem como objectivo a implementação de sistemas difusos com inferência do tipo Takagi-Sugeno e é constituída por uma rede neuronal com cinco camadas. Relativamente às funções de pertença das variáveis de entrada o seu mapeamento é realizado na primeira camada escondida da rede, de seguida, com o operador norma-T calcula-se os antecedentes das regras na segunda camada. A terceira camada normaliza os graus de pertença das entradas, onde na camada 4 os consequentes das regras são determinados. Por fim, a última camada, a camada de saída determina a saída global como a soma de todas as entradas desta camada. Esta arquitectura utiliza um algoritmo de aprendizagem de retropropagação para determinar as funções de pertença dos antecedentes, e utiliza o método dos mínimos quadráticos para calcular as funções de pertença dos consequentes Jang 1993.

Para perceber estas arquitecturas mais aprofundadamente e as suas aplicações, existe uma revisão bibliográfica mais vasta e profunda realizada em 2000 por Abraham Abraham e Nath 2000, onde se pode analisar em maior pormenor as arquitecturas apresentadas e os algoritmos de aprendizagem.

Devido ao facto do controlador a desenvolver nesta dissertação ser semelhante à arquitectura (ANFIS), esta arquitectura irá ser abordada de uma forma mais detalhada no capítulo três.

Capítulo 3

Projecto de Controladores

3.1 Introdução

Neste capítulo vai-se abordar o projecto de controladores difusos e neuro difusos, que estão na base do desenvolvimento do protótipo a implementar nesta dissertação. Vai-se também abordar a estabilidade de sistemas dinâmicos de forma a poder garantir a estabilidade do protótipo a desenvolver. Com este fim vai-se utilizar o algoritmo de LMI's de forma a garantir a estabilidade do sistema em anel fechado.

A arquitectura a implementar é apresentada neste capítulo de uma forma generalista. Esta arquitectura tem como base arquitectura neuro difusa ANFIS razão pela qual esta também será descrita neste capítulo. A arquitectura a implementar é composta pelo modelo neuro difuso proposto. Apresenta-se também o algoritmo de treino do modelo neuro difuso proposto bem como a arquitectura do controlador neuro difuso proposto.

3.2 Controladores Difusos e Neuro Difusos

O projecto de controladores difusos, com inferência de Takagi-Sugeno assentam na utilização de um procedimento de compensação distribuída paralela *Parallel Distributed Compensation* (PDC). Este tipo de abordagem foi apresentada por Sugeno e Kang em 1986 Sugeno e Kang 1986 e melhorada por Tanaka e Sugeno em 1982 Tanaka e Sugeno 1992 introduzindo a análise de estabilidade do sistema.

Para projectar um controlador difuso e por sua vez a utilização do PDC é necessário a construção do modelo difuso T-S do sistema a ser controlado. O controlador difuso apresentará a mesma estrutura do modelo difuso em relação à parte dos antecedentes e a base de regras será a mesma que a do modelo sendo a alteração na componente dos consequentes. O controlador difuso a implementar a partir do PDC para o modelo difuso

que se encontra representado pela equação 2.2, apresenta a seguinte estrutura 3.1:

$$\begin{gathered}\mathbf{R}_i: \\ \textbf{IF } z_1(k) \textit{ is } M_{i1} \textit{ and } ... \textit{ and } z_p(k)) \textit{ is } M_{ip}, \\ \textbf{THEN } \boldsymbol{u}(k) = \boldsymbol{K}_i \boldsymbol{x}(k)\end{gathered} \tag{3.1}$$

Analisando a equação 3.1 pode-se concluir que cada regra $\mathbf{R}_i$ apresenta um ganho linear $\mathbf{K}_i$ que no caso desta equação representa um controlador por retroacção de variáveis de estado. No entanto outro tipo de controladores podem ser projectados.

Assim sendo a lei de controlo difuso resultante é 3.2:

$$\mathbf{u}(k) = \sum_{i=1}^{r} h_i(\mathbf{z}(k))\mathbf{K}_i\mathbf{x}(k)) \tag{3.2}$$

com $h_i(\mathbf{z}(t))$ dado pela equação 2.9.

Substituindo a equação 3.2 na equação 2.5, o sistema em anel fechado fica descrito por 3.3:

$$\dot{\boldsymbol{x}}(k+1) = \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(\mathbf{z}(k))h_j(\mathbf{z}(k))(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j)\boldsymbol{x}(k) \tag{3.3}$$

O projecto de controlador difuso via PDC tem como objectivo determinar os ganhos de realimentação $\mathbf{K}_i$. Estes controladores são construidos através de estruturas de projecto local, representadas pelos modelos lineares locais. Os ganhos de retroacção devem ser calculados em condições de projecto global de forma a garantir a estabilidade e o desempenho do controlador global Tanaka e Wang 2001.

De forma a garantir a estabilidade global do sistema costuma-se utilizar o conceito de estabilidade quadrática. Assim sendo, a condição de estabilidade global é formulada em termos de uma função de Lyapunov do tipo $V(x(t)) = x(t)^T Px(t)$, em que $P = P^T$ e $P > 0$. Tal permite concluir que para calcular os ganhos de realimentação é necessário encontrar uma matriz P que satisfaça a função de Lyapunov de forma a garantir a estabilidade global do sistema difuso.

Para compreender melhor as questões associadas à estabilidade de sistemas dinâmicos apresenta-se na secção seguinte alguns conceitos fundamentais para a construção de controladores, entre eles de referir a análise de estabilidade de sistemas dinâmicos e por sua vez o projecto de controladores de sistemas neuro-difusos.

3.3 Estabilidade de Sistemas Dinâmicos

Os sistemas dinâmicos podem ser representados por um conjunto de equações diferenciais em tempo (contínuo) ou equações às diferenças caso se esteja a representar o sistema em tempo discreto 3.4 e 3.5:

$$\dot{x} = f_c(x,t) \qquad continuo \tag{3.4}$$

$$x_{k+1} = f_k(x_k, k) \qquad discreto \tag{3.5}$$

sendo que $x \in \mathbb{R}^n$ é o vector de estados, e $f_c : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$ e $f_d : \mathbb{R}^n \times \mathbb{Z}_+ \to \mathbb{R}^n$ são funções não lineares ou chamadas campos vectoriais e por sua vez uma trajetória do sistema é um conjunto de valores de $x(t)$ ou x_k, partindo das condições iniciais $x(0)$ e x_0. De seguida vai-se restringir os conceitos aos sistemas contínuos apesar destes também serem válidos para sistemas discretos.

O sistema é dito autónomo quando a equação 3.5 não depende do tempo,

$$\dot{x} = f_c(x) \tag{3.6}$$

caso contrário o sistema diz-se não autónomo.

Os pontos espaciais pertencentes ao campo vectorial f_c são designados por pontos de equilíbrio. Um ponto de equilíbrio, ou ponto fixo no discreto, x^* é designado desta forma se existir uma trajetória que permaneça em x^* Slotine e Li 1991. Isto implica que:

$$\dot{x}(t) = 0 \qquad continuo \tag{3.7}$$

$$x_{k+1} = x_k \qquad discreto \tag{3.8}$$

3.3.1 Estabilidade no Sentido de Lyapunov

Uma das questões mais importantes referentes a sistemas não lineares é a questão da estabilidade Jury 1996, e um dos métodos mais utilizados foi proposto por Lyapunov LaSalle e Lefschetz 1961,Liapunov e Fuller 1992.

Quando um sistema é estável a sua trajectória no espaço de estados permanece próxima de um ponto de equilíbrio caso as condições iniciais sejam próximas desse ponto, ou quando o seu comportamento resista a perturbações.

Definição (Estabilidade no sentido de Lyapunov): *O ponto de equilíbrio $x^* = 0$ é estável no sentido de Lyapunov em t_0 se $\forall \rho > 0$ existir um escalar positivo $r(\rho, t_0)$ tal que :*

$$||x(t_0))|| < r \Rightarrow ||x(t_0))|| < \rho \quad \forall t > t_0 \tag{3.9}$$

caso contrário x^*será instável.

Definição (Atractividade segundo Lyaponov): *O ponto de equilíbrio x^* é atractivo se $\exists\ d > 0$ tal que :*

$$||x(0) - x^*|| \leq d \Rightarrow \lim_{t \to \infty} ||x(t) - x^*|| \to 0 \tag{3.10}$$

Um ponto designa-se atractivo quando existe uma bola de raio d, centrada no ponto x^* tal que trajectórias que partam das condições iniciais partam dessa região para o ponto

de equilíbrio. Pode-se assim concluir que a estabilidade não implica atractividade e vice-versa.

Definição (Estabilidade assimptótica segundo Lyapunov): *O ponto de equilíbrio O é assimptoticamente estável em t_0 se:*

- *Se for estável*
- $\exists r(t_0) > 0 : \|x(t_0)\| < r(t_0) \Rightarrow \|x(t)\| \to 0, t \to \infty$

Este conceito implica a existência de uma sub-região de $\mathbb{R}^n$ atractiva para qualquer instante inicial t_0. A extensão desta região e a velocidade de convergência podem depender do instante inicial t_0.

Definição (Estabilidade Global assimptoticamente estável segundo Lyapunov): *O ponto de equilíbrio O é global assimptoticamente estável para* $\forall\ x(t_0) \in \mathbb{R}^n, \lim_{t\to\infty} x(t) \to O$.

Um sistema é globalmente assimptoticamente estável quando a sua região de atracção é todo o domínio de f_c. Um dos factos que leva a esta definição ser um requisito importante para projectar um controlador prende-se com o facto destas definições dependerem das condições iniciais, e por sua vez o controlador ter a possibilidade de partir de quaisquer condições iniciais.

3.3.2 Método Direto de Lyapunov

Este método baseia-se no estudo de uma função escalar invariante no tempo que deverá apresentar propriedades particulares para que o sistema seja estável no sentido de Lyapunov.

Definição: *Uma função escalar $v(x,t)$ é localmente positiva definida se $v(0,t) = 0$ e existir uma função invariante definida positiva $v_0(x)$ tal que:*

$$v(x,t) \geq v_0(x), \forall t \geq 0 \tag{3.11}$$

Definição: *Uma função escalar $v(x,t)$ é decrescente se $v(0,t) = 0$ e existir uma função invariante definida positiva $\bar{v}(x)$ tal que:*

$$v(x,t) \leq \bar{v}(x), \forall t \geq 0 \tag{3.12}$$

Teorema 1: *(Teorema de Lyapunov para sistemas não autónomos). Se numa bola B centrada no ponto de equilíbrio O existir uma função escalar $v(x,t)$ com derivadas parciais contínuas tal que:*

1. *$v(x,t)$ é definida positiva.*
2. *$\frac{\partial v}{\partial t}$ é semi-definida negativa, então o ponto de equilíbrio O é estável no sentido de Lyapunov.*

3. *Se* $v(x,t)$ é decrescente, então a origem é estável e se $\dot{v}$ é definida negativa então o ponto de equilíbrio O é assimptoticamente estável.

4. Se B coincidir com $\mathbb{R}^n$ e todas as condições anteriores forem satisfeitas, então o ponto de equilíbrio diz-se global assimptoticamente estável.

Infelizmente este método não determina uma função de Lyapunov, mas dado essa função determina se o sistema é estável.

3.4 LMI-Linear Matrix Inequality

As LMI's surgiram como ferramentas muito poderosas para lidar com problemas de controlo que se apresentam difíceis ou mesmo impossíveis de resolver de uma forma analítica.

Apesar das LMI's datarem já dos anos 40 do século passado, com maior ênfase no controlo a reportar aos anos 60 do século passado com o trabalho de Kalman Kalman 1963, Popov Popov 1961 e Willems Willems 1971, somente nas últimas décadas foram desenvolvidas técnicas numéricas para resolução das LMI's de uma forma prática e muito eficiente como por exemplo o trabalho de Nesterov e Nemirovskii de 1994 "Interior Point Polynomial Time Methods in Convex Programming" 2004. A *LMI Control toolbox* do Matlab Gahinet et al. 1994 é baseada nesse mesmo algoritmo e oferece uma interface gráfica e extenso suporte para as aplicações de controlo.

Actualmente existem muitos pacotes de software comercial e não comercial que permitem uma codificação simples da maior parte dos problemas de LMI's e fornecem ferramentas para resolução de problemas de controlo de uma forma eficiente.

Como já foi referido na secção 3.3, as LMI's surgiram como uma abordagem para o cálculo das funções de Lyapunov, de forma a se poder investigar a estabilidade do sistema. Analisando as equações 3.13 e 3.14 e utilizando uma abordagem LMI's, estas calculam a matriz P permitindo a análise da estabilidade em anel aberto. De seguida é necessário analisar a estabilidade em anel fechado recorrendo às LMI's para a resolução do método de Lyapunov. Com as equações que descrevem o modelo T-S mais as equações do controlador pode-se definir uma equação geral que define o sistema em anel fechado. De seguida ao executar o algoritmo LMI's pode-se adquirir os ganhos do controlador que permitem que o sistema em anel fechado seja estável. Com este objectivo pode-se utilizar diversas funções do Matlab para a implementação de LMI's de forma a calcular os ganhos do controlador que garantem a estabilidade do sistema em anel fechado.

Um dos problemas na utilização das LMI's prende-se com o facto desta abordagem não calcular os melhores ganhos mas sim calcular uma região onde estes garantem a estabilidade do sistema. Deste modo para se calcular os melhores ganhos do controlador terá de se utilizar uma abordagem de LMI's e uma métrica para a escolha dos melhores ganhos podendo esta última ser calculada através de algoritmos optimização para minimizar o erro de seguimento, ou a variação da acção de controlo.

3.5 Estabilidade de Sistemas Neuro-Difusos

Uma das abordagens para investigar a estabilidade de sistemas neuro-difusos, foi desenvolvida por Tanaka e Sugeno, Tanaka e Sugeno 1992, onde se pode concluir o seguinte teorema:

Teorema 2: *Um sistema T-S é assintoticamente estável se existir uma matriz* $P = P' \rightarrow 0$ *e* $P > 0$ *definida positiva, tal que :*

$$A_j' P A_j - P \rightarrow 0, \qquad j \in \mathbb{R} \qquad CFS \tag{3.13}$$

$$A_j' P A_j - P \rightarrow 0, \qquad j \in \mathbb{R} \qquad DFS \tag{3.14}$$

Esta metodologia infelizmente não é sistemática e geral para todos os sistemas, obrigando assim vários testes na variável P para obtenção da matriz. Devido a este facto é possível encontrar na bibliografia diversos trabalhos, Tanaka e Sugeno 1992; Tanaka e Sano 1993, que utilizam métodos heurísticos para determinar as matrizes da função de Lyapunov. Já Kawamoto desenvolveu um método analítico para sistemas T-S de segunda ordem Kawamoto et al. 1992.

Este facto fez com que mais tarde com a evolução de algoritmos mais eficientes, desse origem ao desenvolvimento da abordagem das LMI's aplicadas em controlo.

Utilizando então o contexto das LMI's para calcular os ganhos $\mathbf{K}_i$ podem ser definidas como em Tanaka e Wang Tanaka e Wang 2001:

Achar uma matriz $X = X^T, X > 0$ e $M_i (i = 1, 2, ..., r)$, *que satisfaça as seguintes LMI's*

$$-XA_i^T - A_i X + M_i^T B_i^T + B_i M_i > 0; \forall i = 1, 2, ..., r \tag{3.15}$$

$$-XA_i^T - A_i X - XA_j^T - A_j X + M_j^T B_i^T + B_i M_j + M_i^T B_j^T + B_j M_i \geq 0; \forall i < j, i, j = 1, 2, ..., r \tag{3.16}$$

em que $X = P^{-1}$ *e* $M_i = K_i X$ *, r é o número de regras do sistema difuso, de forma que o equilíbrio do sistema T-S é globalmente assintóticamente estável.* Outra possibilidade consiste na restrição da região LMI's.

3.6 Projecto de Controladores Neuro-Difusos

Com vista ao projecto de controladores neuro-difusos vai-se abordar em maior detalhe a arquitectura ANFIS já referida no capítulo 2. Tal deve-se ao facto de esta ser semelhante à arquitectura que se irá utilizar para a construção do controlador neuro-difuso a implementar neste trabalho.

Como já foi anteriormente referido a arquitectura ANFIS implementa sistemas difusos de inferência difusa do tipo Takagi-Sugeno, sendo constituída por uma rede neuronal

do tipo *feedforward* de cinco camadas. Este sistema com algumas adaptações pode representar diferentes inferências difusas e diferentes bases de regras Lee 1990a; Lee 1990b. Este facto leva à centralização da arquitectura ANFIS num tipo especifico de inferência, onde a saída de cada regra consiste numa combinação linear de variáveis de entrada mais um termo constante, e a saída final do controlador consiste na média ponderada de cada regra disparada.

Assim sendo, para melhor análise da arquitectura ANFIS apresenta-se na Figura 3.1 a esquematização da mesma:

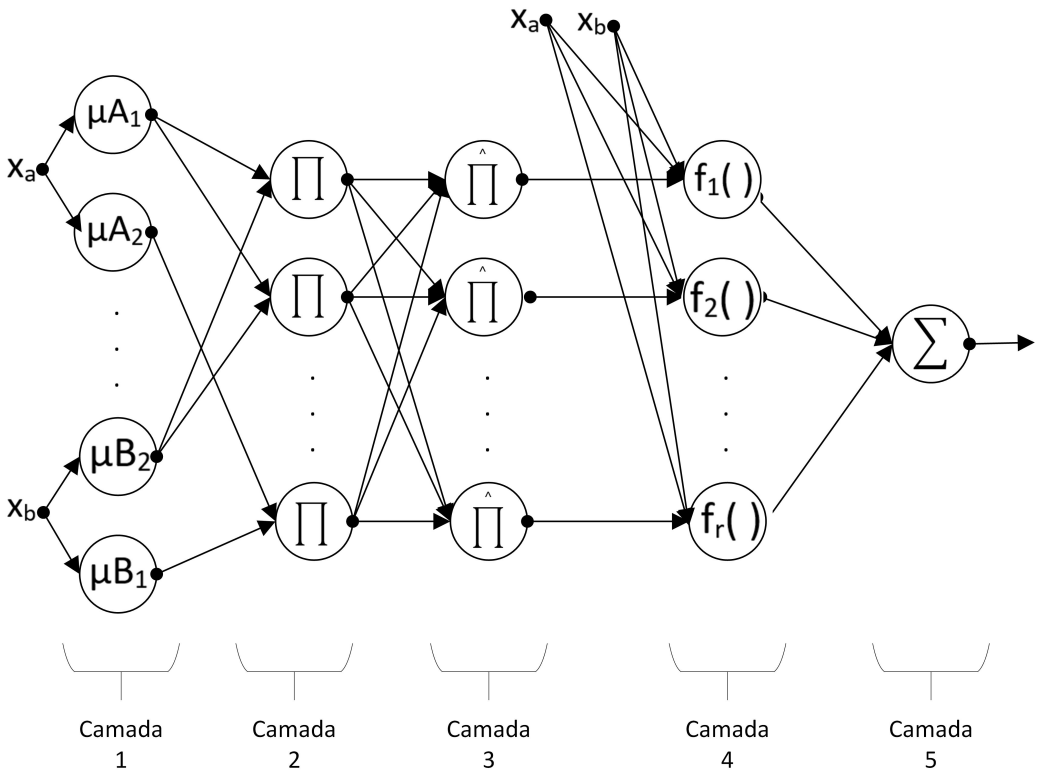

Figura 3.1: Arquitectura ANFIS.

Através da observação da Figura 3.1, vai-se analisar a arquitectura ANFIS por camadas.

Na primeira camada todos os neurónios i têm uma função de activação do tipo:

$$O_i^1 = \mu A_i(x_a) \tag{3.17}$$

onde x_a e x_b são os valores de entrada dos neurónios i e por sua vez A_i e B_i corresponde à etiqueta linguística, o que permite concluir que O_i^1 corresponde às funções de pertença do sistema difuso, representando o grau de quanto a entrada x_a satisfaz a etiqueta linguística A_i e quanto a entrada x_b satisfaz a etiqueta linguística B_i . Normalmente

as funções de pertença $\mu A_i(x)$ têm um máximo em 1 e um mínimo em 0, o que faz com que possam tomar a seguinte forma:

$$\mu A_i(x) = \frac{1}{1 + \left[\left(\frac{x-c_i}{a_i}\right)^2\right] b_i} \tag{3.18}$$

ou

$$\mu A_i(x) = exp\left\{-\left(\frac{x-c_i}{a_i}\right)^2\right\} \tag{3.19}$$

Assim com os parâmetros a_i, b_i e c_i, denominados de parâmetros antecedentes, podemos alterar a forma das funções de pertença, permitindo melhorar o desempenho do controlador ANFIS.

Já o objectivo dos neurónios da segunda camada consiste em enviar a multiplicação dos sinais de entrada e enviar o resultado desta para a camada seguinte. Esta multiplicação é representada por:

$$w_i = \mu A_i(x) \times \mu B_i(y), \qquad i = 1, 2, ..., n \tag{3.20}$$

sendo que w_i representa o grau de de disparo de cada regra. Também podem ser utilizados outros operadores T-norma que desempenham o mesmo tipo de função (AND), implementados como função de activação nesta camada.

Cada neurónio, na camada três, calcula a razão do grau de disparo da regra i, em relação à soma dos graus de disparo de todas as regras de acordo com:

$$\overline{w_i} = \frac{w_i}{w_1 + w_2 + ... + w_n}, \qquad i = 1, 2, ..., n \tag{3.21}$$

A saída $\overline{w_i}$ é denominada de grau de disparo normalizado.

Todos os neurónios i, da camada quatro têm a seguinte função de activação:

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{3.22}$$

onde f_i, representa o consequente de cada regra, $\overline{w_i}$ corresponde a saída da camada três, e p_i, q_i e r_i, correspondem ao conjunto de parâmetros dos consequentes.

Na camada cinco calcula-se a saída como soma de todos os sinais de entrada, ou seja:

$$O_1^5 = \sum_{i=1}^{n} O_i^4 = \frac{\sum_{i=1}^{n} w_i f_i}{\sum_{i=1}^{n} w_i} \tag{3.23}$$

Constrói-se assim uma rede adaptativa *feedforward* cuja funcionalidade é equivalente a um sistema de inferência difusa do tipo Takagi-Sugeno com regras do tipo *if-then* tal como podem ser consultadas em Takagi e Sugeno 1983.

3.7 Arquitectura Neuro Difusa Proposta

O protótipo a desenvolver tem como base a construção de uma arquitectura neuro difusa, e como já foi anteriormente referido esta será muito semelhante à arquitectura neuro difusa ANFIS, com diferenças no número de camadas da rede neuronal, e nos consequentes da inferência difusa. Este apresenta também algumas alterações no treino da rede neuronal de forma a evitar a ocorrência do sobreajuste do protótipo em relação ao conjunto de treino da rede neuronal.

3.7.1 Modelo Neuro Difuso Proposto

O modelo neuro difuso proposto tem como base o modelo neuro difuso ANFIS, mas com a adição de uma camada na rede neuronal e alteração dos modelos lineares que passam a estar representados no espaço de estados. Assim, de forma a compreender melhor as alterações realizadas apresenta-se a Figura 3.2.

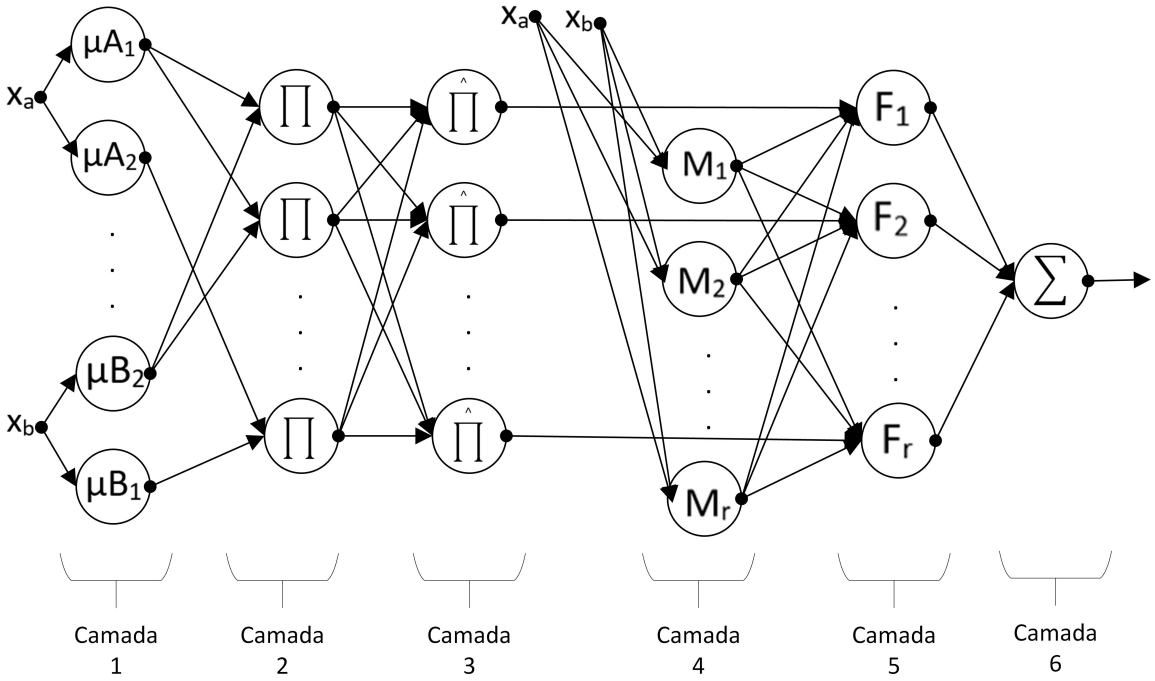

Figura 3.2: Arquitectura Proposta para o Modelo Neuro Difuso.

Analisando a Figura 3.2 da arquitectura proposta e a Figura 3.1 referente à arquitectura ANFIS pode-se constatar que a estrutura da camada 1 à camada 3 da rede neuronal

apresenta-se de igual forma sendo esta representada pelas mesmas equações que na arquitectura ANFIS. Em relação à camada 4, à camada 5 e à camada 6 o mesmo não se verifica, uma vez que foi adicionada uma nova camada e as equações que as descrevem são diferentes. Assim sendo pode-se verificar na Figura 3.2 que na camada 4 estão presentes os modelos lineares no espaço de estados, pertencentes aos consequentes da inferência difusa, sendo os modelos lineares, associados à camada 4, representados pela equação genérica (3.24).

$$O_i^4 = \begin{cases} \boldsymbol{x}(t+1) = \boldsymbol{A}_i\boldsymbol{x}(t) + \boldsymbol{B}_i\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}_i\boldsymbol{x}(t) \end{cases} \quad i = 1, 2, ..., r. \tag{3.24}$$

Observando a equação (3.24) pode-se verificar que A_i, B_i e C_i são as matrizes que descrevem o modelo linear no espaço de estados e que $x(t)$ corresponde às variáveis de estado e por sua vez $u(t)$ e $y(t)$ correspondem à entrada e saída do modelo respectivamente.

Na camada 5 tem-se a associação das regras disparadas, presentes na camada 3 da rede neuronal, com os consequentes da inferência difusa presentes na camada 4. A camada 5 é representada pela equação 3.25:

$$F_i = O_i^5 = \sum_{i=1}^{r} O_i^4 w_i^4 \quad i = 1, 2, ..., r. \tag{3.25}$$

Pode-se constatar pela equação 3.25, que na camada 5 o O_r^4 corresponde aos modelos lineares escolhidos e por sua vez w_r^4 corresponde a um peso que associa o modelo linear às regras. Assim sendo, a multiplicação $O_r^4 w_r^4$ representa o consequente de cada regra e $\overline{w_i}$ é a saída da camada três correspondendo ao grau de disparo normalizado de cada regra.

Na camada 6 calcula-se a saída da rede neuronal como a soma de todos os sinais de entrada, ou seja :

$$O_1^6 = \sum_{i=1}^{n} O_i^5 \tag{3.26}$$

Observando a equação 3.26 pode-se concluir que a saída do modelo não linear corresponde ao somatório da saída dos modelos lineares ponderados pelos pesos da camada 5, e ponderados pelos graus de disparo normalizados da camada 3.

3.7.2 Treino do Modelo Neuro Difuso Proposto

Neste capítulo vai-se abordar o treino do modelo neuro difuso proposto de modo a se adaptar à instalação.

Analisando a Figura 3.2 pode-se constatar que as camadas onde o treino irá incidir será nos pesos da camada 1 e nos pesos da camada 4, sendo todos os outros pesos das outras camadas igual à unidade. Este treino aplica-se nestas camadas pois estas são responsáveis pelos antecedentes e pelos consequentes da inferência difusa.

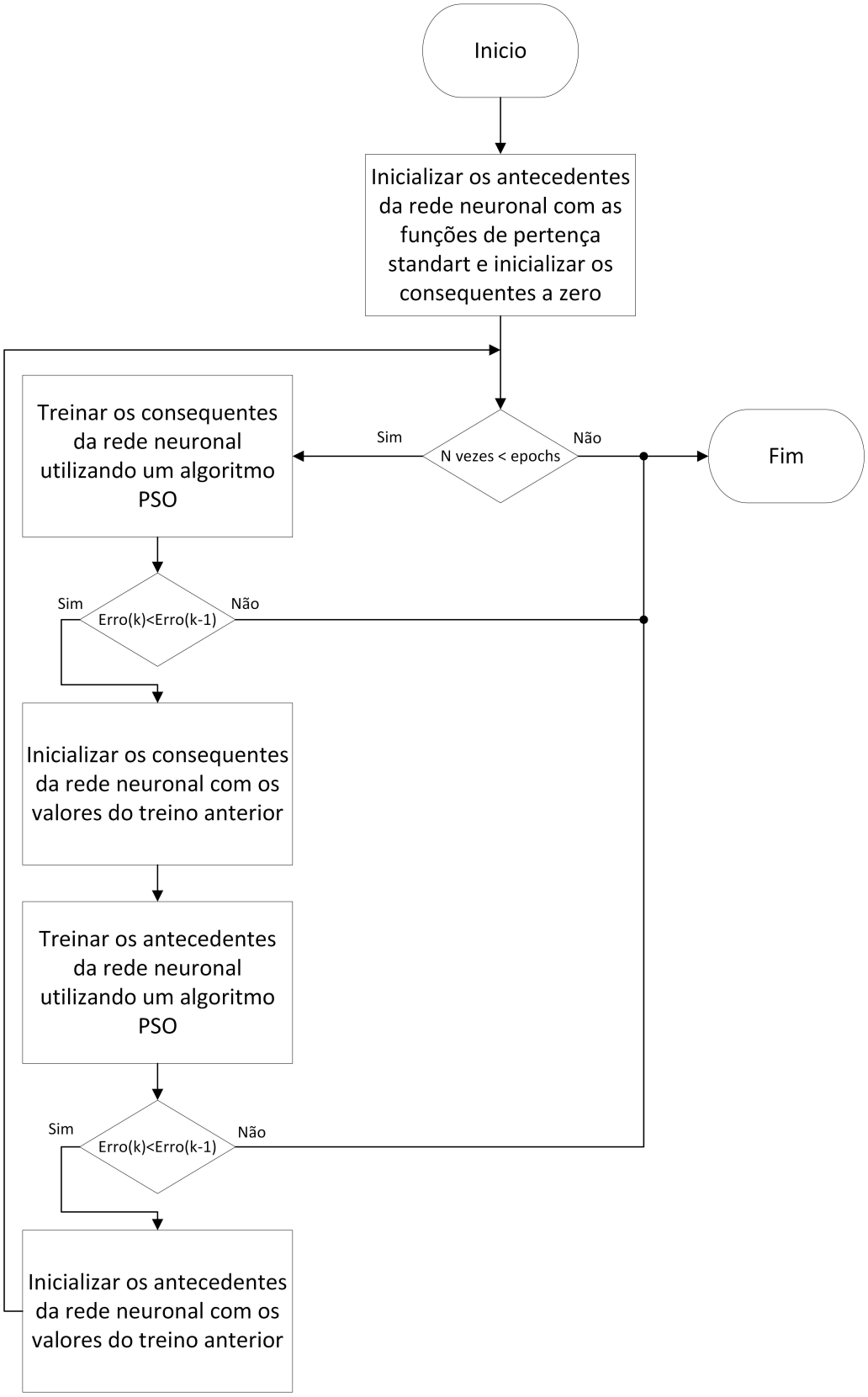

Figura 3.3: Algoritmo de treino da arquitectura proposta

Devido ao facto desta arquitectura ser bastante semelhante à arquitectura ANFIS pensou-se na utilização do algoritmo base para o treino desta arquitectura. Este algoritmo de base utiliza o gradiente descendente com o algoritmo *backpropagation* para optimizar os antecedentes e utiliza o *Least Square Estimation* (LSE) para os consequentes. Este algoritmo base veio a constatar-se obsoleto por, Karaboga e Kaya Karaboga e Kaya 2018. Assim sendo, segundo os mesmos autores, um dos melhores algoritmos a implementar seria o algoritmo de treino proposto apresentado na Figura 3.3.

O algoritmo da Figura 3.3, utiliza um algoritmo *Particle Swarm Optimization* (PSO) com o algoritmo *backpropagation* de forma a optimizar as funções de pertença, e utiliza um algoritmo PSO de forma a optimizar os pesos que associam os modelos lineares em espaço de estados às regras. Este algoritmo revela-se melhor que o algoritmo base do ANFIS pois anula o *overfiting* aos dados de treino.

3.7.3 Controlador Neuro Difuso Proposto

O controlador neuro difuso tem por base a arquitectura do modelo neuro difuso proposto, com excepção da camada 4. Tal deve-se ao facto do objectivo do controlador ser o de calcular a acção de controlo a aplicar ao sistema e não emular o sistema real nem calcular a saída do sistema.

Com este fim pode-se concluir que o controlador neuro difuso proposto na camada 4 será representado por controladores e não por modelos lineares em espaço de estados.

Deste modo apresenta-se na Figura 3.4 a arquitectura do controlador neuro difuso proposto:

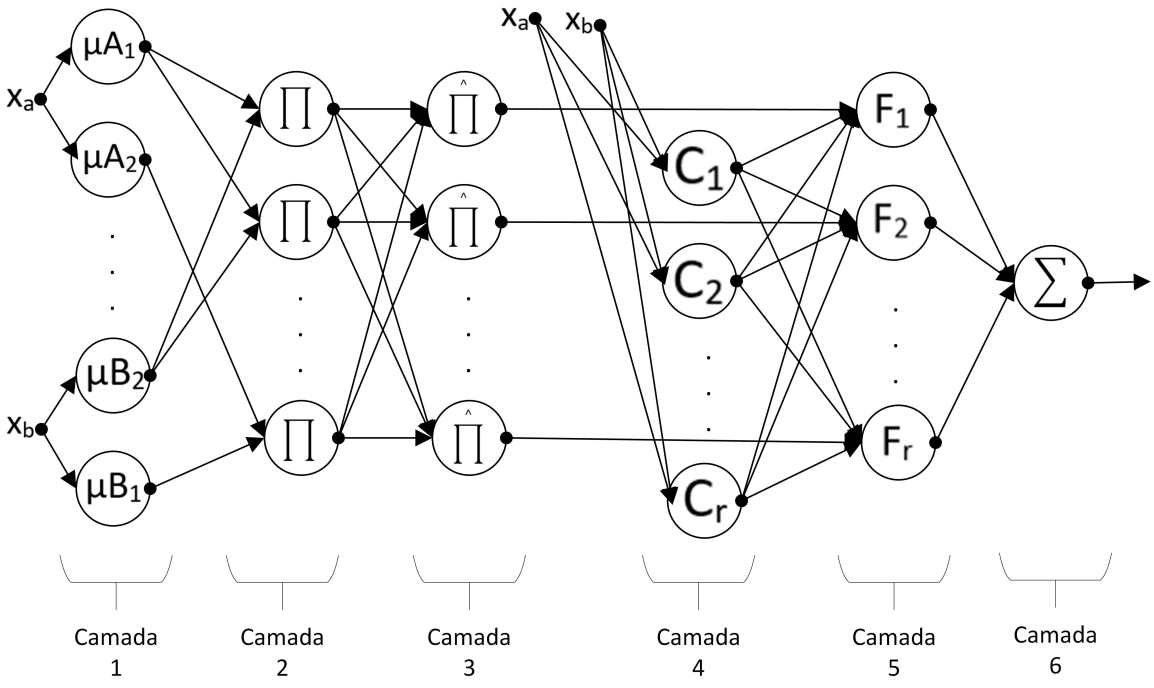

Figura 3.4: Arquitectura do Controlador Neuro Difuso Proposto.

Observando a arquitectura do controlador proposto, Figura 3.4, verifica-se que esta arquitectura se apresenta semelhante à arquitectura do modelo neuro difuso proposto com excepção da camada 4, como foi anteriormente referido. Por observação da camada 4 é possível constatar que o controlador neuro difuso proposto é composto por r controladores. Estes controladores podem ser de diferentes tipologias.

Apresenta-se na Figura 3.5 a tipologia escolhida para a arquitectura do controlador proposto.

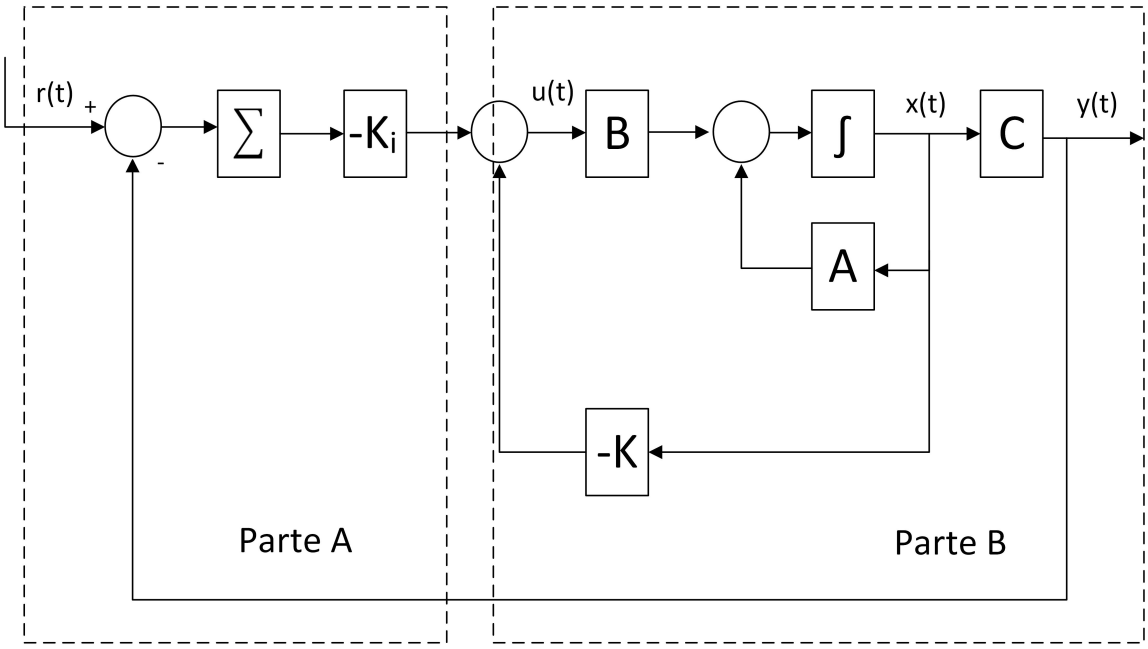

Figura 3.5: Controlador por retroacção de variáveis de estado

Com base na Figura 3.5 pode-se constatar a tipologia escolhida para a arquitectura proposta. Esta escolha tem como base a tipologia de controlo por retroacção de variáveis de estado com efeito integral.

Tal opção advém da utilização dos modelos lineares em espaço de estados na construção da arquitectura do modelo neuro difuso.

Os controladores por retroacção de variáveis de estado com efeito integral calculam a acção de controlo em duas componentes:

1. Acção de controlo integral

2. Acção de controlo interna

A acção de controlo integral é obtida pelo cálculo do erro acumulado multiplicado pelo ganho K integral. Na Figura 3.5 pode-se observar o cálculo desta componente da

acção de controlo analisando a parte A da imagem. Já a acção de controlo interna ao sistema é obtida através das matrizes A e B, que permitem calcular o valor das variáveis de estado, sendo estas sujeitas a retroacção e multiplicadas por um ganho K interno, como se pode comprovar pela parte B da Figura 3.5.

Estes ganhos de K com os blocos de retroacção permitem fechar o anel de controlo. A obtenção destes toma uma grande importância de forma a garantir o desempenho e estabilidade do controlador. Assim sendo, devido ao facto da arquitectura de controlo neuro difusa conter r controladores por retroacção de variáveis de estado não é possível utilizar os métodos tradicionais de sintonização deste tipo de controladores. Os métodos tradicionais impossibilitam a conclusão da estabilidade de controladores neuro difusos, pois estes apresentam o cálculo da acção de controlo como a soma ponderada das acções de controlo dos controladores por retroacção de variáveis de estado. Assim sendo uma abordagem tradicional permite garantir a estabilidade de cada controlador por retroacção de variáveis de estado e não do sistema global. Desta forma será utilizada uma abordagem em LMI's de forma a sintonizar os ganhos K garantindo a estabilidade do sistema.

Capítulo 4

Implementação

4.1 Introdução

Neste capítulo vai-se abordar a construção do protótipo desenvolvido desde a fase de modelação até a fase de construção do controlador neuro difuso proposto, passando por diversas etapas de implementação. Na Figura 4.1 encontram-se descritas as etapas mais importantes para a implementação do protótipo.

Analisando a Figura 4.1 pode-se constatar que para implementar o protótipo é necessário em primeiro lugar obter os modelos lineares por identificação em linha em determinados pontos de funcionamentos, e por sua vez proceder à validação dos modelos obtidos. Como segunda etapa tem-se a construção do modelo não linear e o respectivo treino da rede neuronal, validando este modelo para a instalação em questão. A terceira etapa tem como objectivo a obtenção dos ganhos de controlo garantindo a estabilidade global do sistema, sendo para tal utilizada uma abordagem baseada em LMI's. A quarta etapa envolve a construção do controlador e do anel de controlo e a sua validação para a instalação. Finalmente a última etapa tem como objectivo a síntese de um sistema anticolagem e a sua validação.

Após a construção do protótipo desenvolvido nesta dissertação vai-se também comparar o controlador desenvolvido com um controlador PID difuso do tipo Mandani, de forma a se poder averiguar o desempenho do controlador projectado. O controlador PID difuso foi implementado utilizando inferência difusa do tipo Mandani *Aplicação de técnicas de controlo óptimo difuso em ambientes distribuídos*. Com este fim recorreu-se a três métricas de desempenho o *Root Mean Square Error* (RMSE), *Mean Square of Control Action Increment* (MSI), e Energia Média associada acção de controlo representadas nas equações 4.1, 4.2 4.3 respectivamente de forma analisar e comparar o desempenho dos controladores:

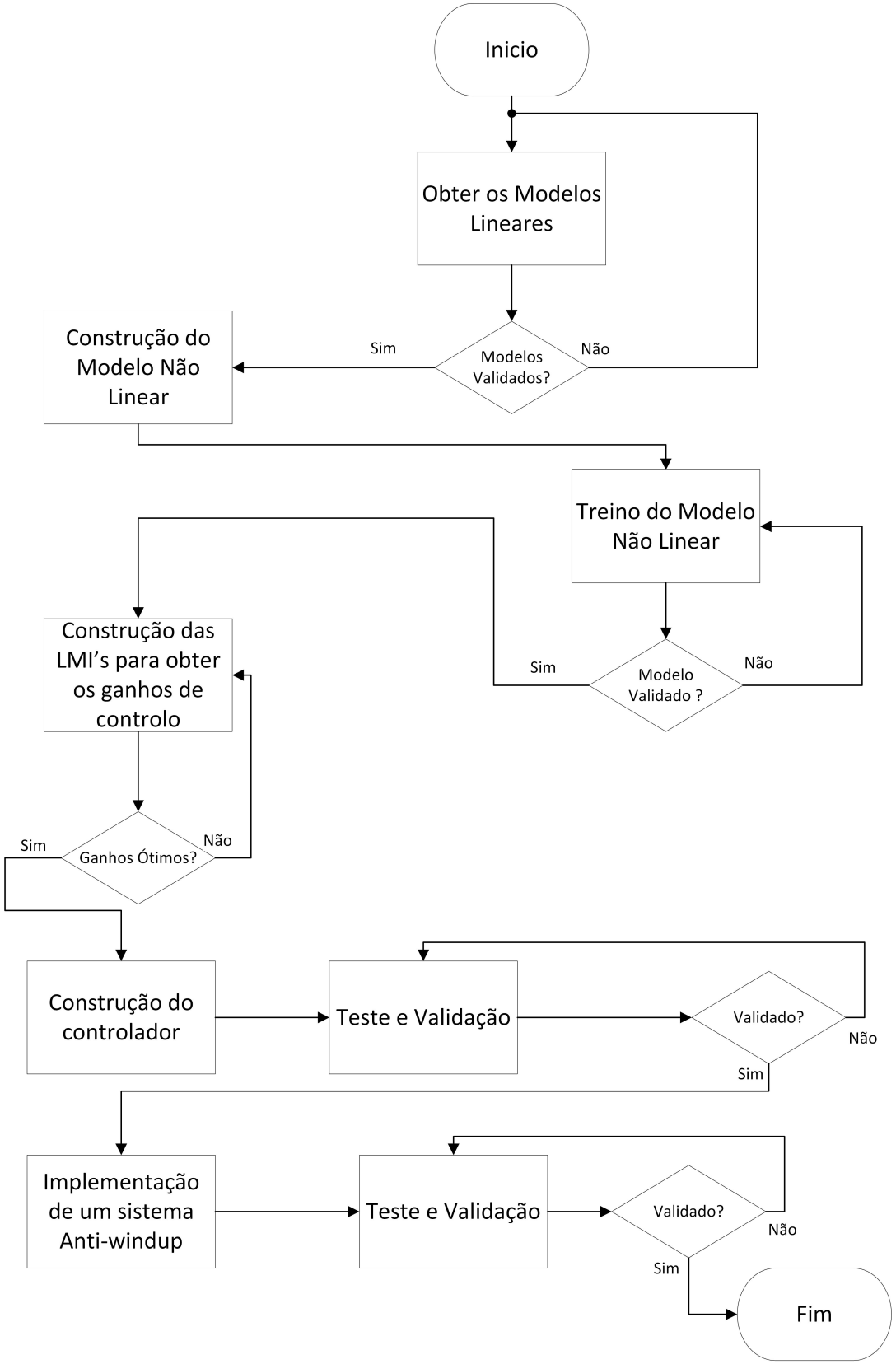

Figura 4.1: Fases de implementação do protótipo

$$RMSE = \sqrt{\frac{\sum_1^k (y(k) - r(k))^2}{k}} \tag{4.1}$$

$$MSI = \frac{\sum_2^k |u(k) - u(k-1)|}{k} \tag{4.2}$$

$$EnergiaMedia = \frac{\sum_1^k (u(k))^2}{k} \tag{4.3}$$

Analisando a equação (4.1), pode-se concluir que esta equação permite calcular o erro que corresponde à raiz do quadrado da diferença do vector de saída $y(k)$ em relação ao vector de referência $r(k)$ a dividir pelo número de amostras.

Observando as equações (4.2) e (4.3), pode-se constatar que na primeira equação relativa ao MSI se calcula o incremento médio da acção de controlo utilizando o vector da acção de controlo $u(n)$ e a segunda equação calcula a energia média a utilizar no controlador, utilizando o mesmo vector.

4.2 Processo AMIRA DTS200

De forma a se testar e validar a arquitectura proposta neste trabalho, utilizou-se o processo Amira DTS200, com o objectivo de controlar os níveis de água dentro dos tanques. Esta instalação baseia-se num sistema auto-suficiente constituído por três tanques de armazenamento de líquidos, comunicantes entre si, através de duas electrobombas, responsáveis pelos caudais de entrada nos reservatórios periféricos (Tanque 1 e Tanque 2), duas válvulas a ligar estes mesmos ao reservatório intermédio e por fim quatro válvulas que ligam a um reservatório exterior. É provido de três sensores de nível, um para cada tanque, com o objectivo de medir a altura de líquido presente em cada tanque. Por sua vez os tanques apresentam uma forma cilíndrica com uma área de secção constante entre eles e estão nivelados todos à mesma altura. As electrobombas apresentam uma alimentação entre -12 a 12 Volts e os sensores de nível uma tensão de alimentação de -10 a 10 Volts.

O sistema é ainda composto por uma placa de aquisição de dados USB DaQ NI-USB6009, o que faz com que seja necessário um conversor de tensão pois a placa de aquisição para os actuadores opera a uma tensão de alimentação compreendida entre 0 e 5 Volts.

Este processo foi o escolhido para validar os resultados, devido ao facto de este ser um sistema não linear e afim e por sua vez ser considerado um sistema comutado pois permite várias configurações possíveis. Escolheu-se a configuração representada na figura 4.2 devido a esta conter não linearidades e não existir a preocupação com a inversão de fluxos nos tanques uma vez que tal não ocorre nesta configuração.

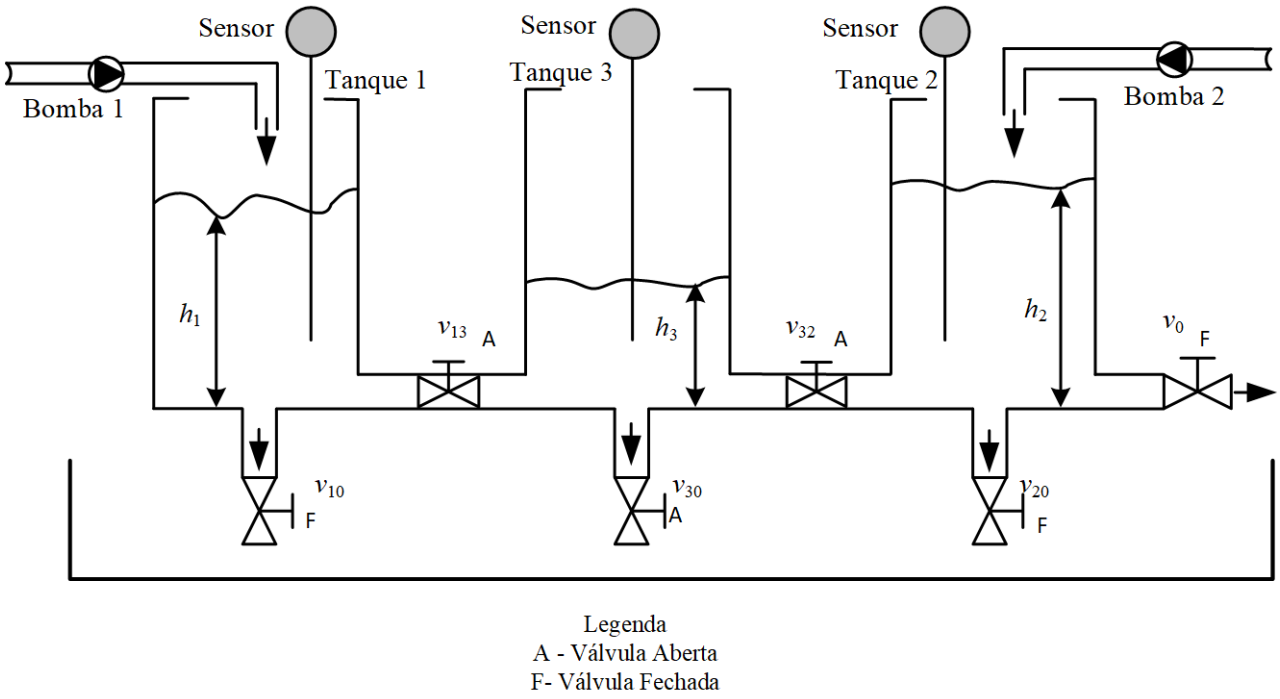

Figura 4.2: Esquema da configuração escolhida do processo Amira DTS200

Analisando a Figura 4.2 pode-se identificar que h_1, h_3 e h_2 correspondem ao nível de cada um dos tanques. As válvulas de comunicação entre os tanques são a v_{13} que liga o tanque 1 ao tanque 3 e a v_{32} que liga o tanque 3 ao tanque 2. As válvulas de descarga de cada tanque estão identificadas como v_{10}, v_{30} e v_{20}, respectivamente, com excepção do tanque 2 que contem mais uma válvula de descarga identificada por v_0.

4.3 Identificação em Diferido

De forma a se construir o modelo não linear através de modelos lineares, optou-se por construir os modelos lineares por identificação em diferido tal como descrito na secção 2.3.

Esta é uma situação que ocorre muitas vezes na indústria por dificuldades associadas à construção do modelo analítico da instalação, o que leva à construção de modelos por identificação em diferido.

Para obter os dados para a construção dos modelos lineares, consideraram-se três zonas distintas que correspondem a diferentes alturas do tanque. A altura do tanque acima de 80% não foi considerada para não estar a operar no limite da instalação de forma a não por as bombas num esforço excessivo. Após definir as três zonas de funcionamento, construiram-se as acções de controlo da bomba um e bomba dois para as três zonas respectivamente. Estas acções de controlo têm a forma de pulsos com duração e intensidades aleatórias ao longo do tempo. Após ter as acções de controlo procedeu-se à excitação da instalação com estas e obtiveram-se as saídas dos tanques para esta excitação obtendo-se

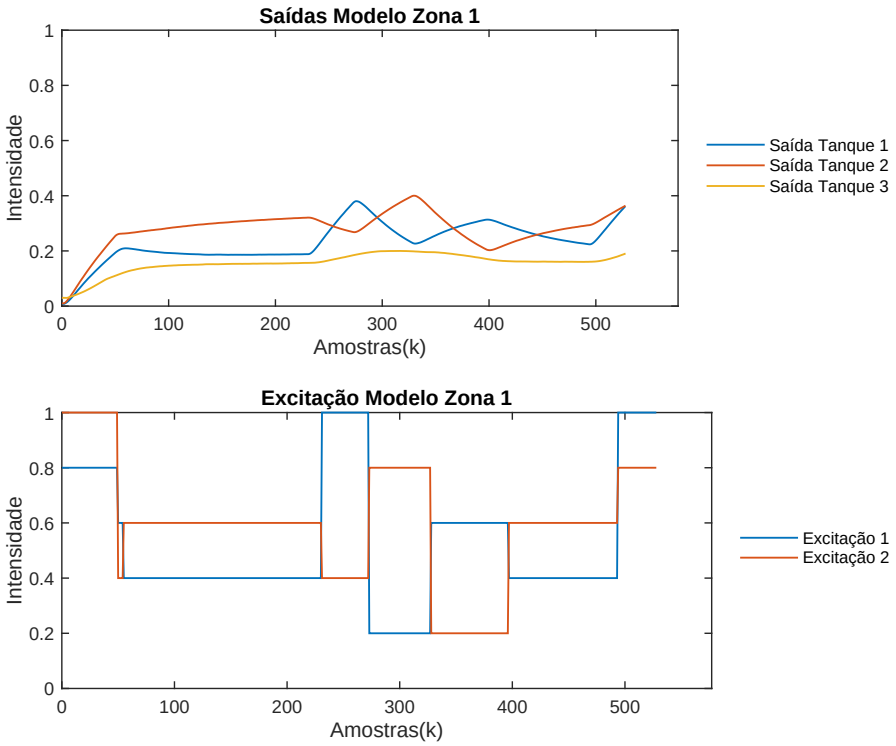

Figura 4.3: Dados para identificação *offline* Zona 1.

assim os dados para a construção dos modelos lineares. Assim sendo, foram retirados dados variando as acções de controlo de forma a manter a saída em diferentes *setpoints* (zonas de funcionamento), representados na Figura 4.3, 4.4 e 4.5.

De acordo com as Figuras 4.3, 4.4, 4.5 pode-se constatar as diferentes zonas escolhidas para construir os três modelos lineares. As três zonas diferentes de funcionamento foram seleccionadas para permitir a construção dos modelos lineares para cada uma destas zonas. Pode-se verificar também a aleatoriedade das acções de controlo. Estas apresentam-se com a aleatoriedade ilustrada nas Figuras com o objectivo dos modelos conterem maior número de dinâmicas do processo. Assim analisando as Figuras 4.3, 4.4, 4.5 podemos constatar as variações das saídas e da excitação do processo para a zona 1 , zona 2 e zona 3, respectivamente.

Após recolher os dados construiram-se os modelos lineares no espaço de estados utilizando a função do matlab n4sid, sendo estes representados pelas expressões 4.4, 4.5 e 4.6. Foram apenas modelados três modelos lineares devido ao facto de a modelação de mais modelos lineares levava a uma complexidade a nível computacional para construir o modelo não linear.

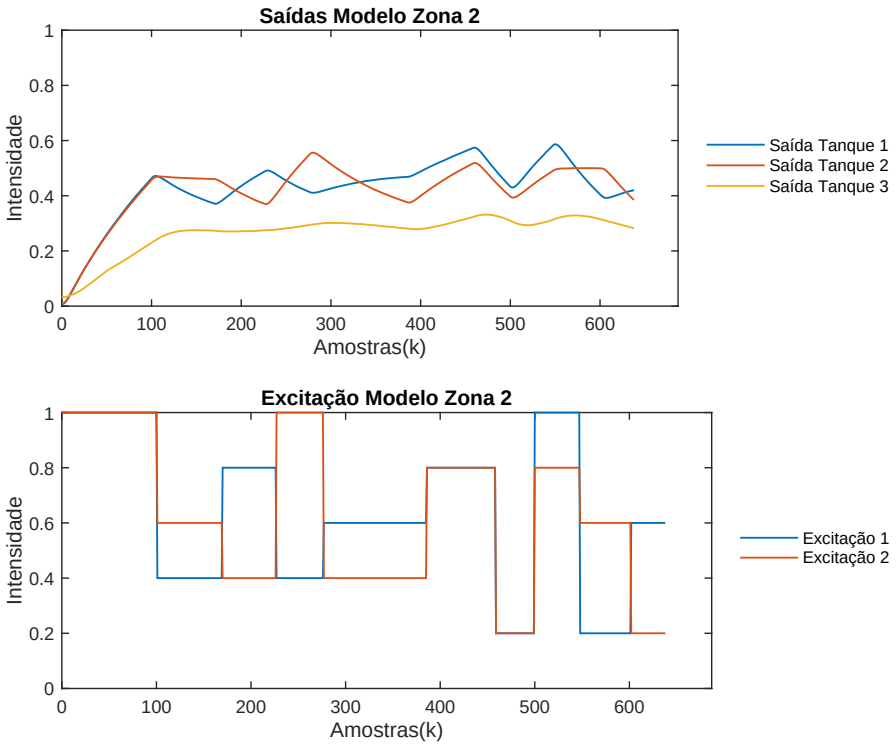

Figura 4.4: Dados para identificação *offline* Zona 2.

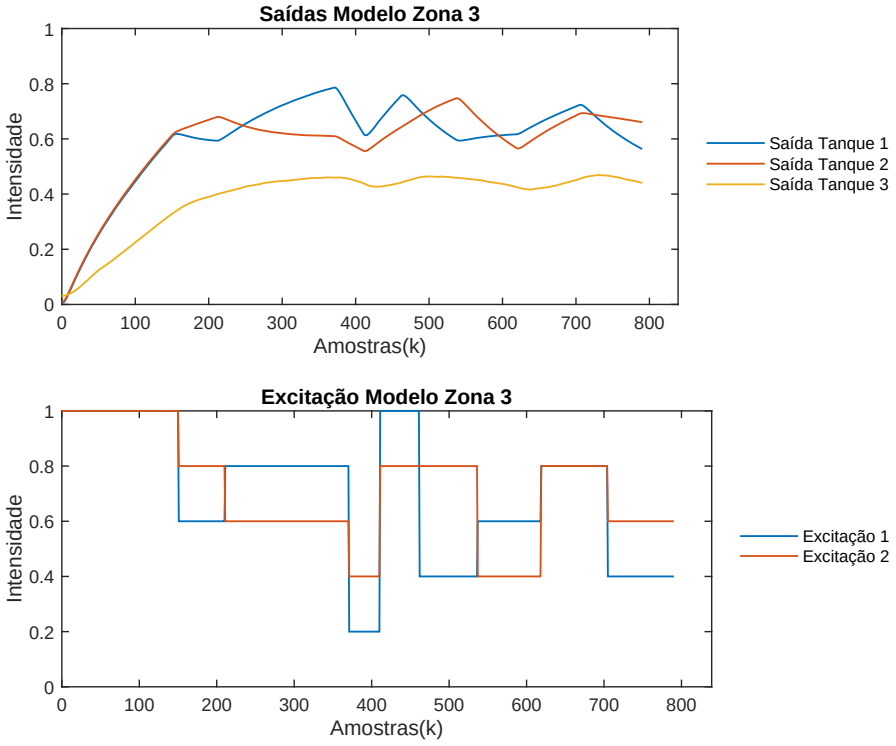

Figura 4.5: Dados para identificação *offline* Zona 3.

$$Modelo1\begin{cases} x(k+1) = \begin{bmatrix} 1.0005 & 0.0056 & -0.0277 \\ 0.0064 & 0.9963 & -0.0235 \\ 0.0101 & 0.0101 & 0.9630 \end{bmatrix} * \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0.0082 & -0.0017 \\ -0.0012 & 0.0075 \\ 0.0007 & 0.0009 \end{bmatrix} * \begin{bmatrix} u_1(k)) \\ u_2(k)) \end{bmatrix} \\ y(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \end{cases} \tag{4.4}$$

$$Modelo2\begin{cases} x(k+1) = \begin{bmatrix} 0.9993 & 0.0004 & -0.0128 \\ 0.0041 & 0.9942 & -0.0109 \\ 0.0090 & 0.0096 & 0.9693 \end{bmatrix} * \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0.0081 & -0.0015 \\ -0.0013 & 0.0080 \\ 0.0004 & 0.0006 \end{bmatrix} * \begin{bmatrix} u_1(k)) \\ u_2(k)) \end{bmatrix} \\ y(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \end{cases} \tag{4.5}$$

$$Modelo3\begin{cases} x(k+1) = \begin{bmatrix} 0.9960 & 0.0007 & -0.0039 \\ 0.0017 & 0.9921 & -0.0006 \\ 0.0058 & 0.0058 & 0.9812 \end{bmatrix} * \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0.0084 & -0.0023 \\ -0.0014 & 0.0080 \\ 0.0004 & 0.0009 \end{bmatrix} * \begin{bmatrix} u_1(k)) \\ u_2(k)) \end{bmatrix} \\ y(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \end{cases} \tag{4.6}$$

Ao utilizar a função n4sid do matlab restringiu-se a matriz C dos modelos lineares de forma a torná-la numa matriz identidade.

Tal procedimento foi feito para evitar a implementação de um observador de estado, que seria desenvolvido através da implementação de um filtro de Kalman, de tal forma que permitisse relacionar a saída real do sistema com os estados. A decisão de utilizar a matriz C dos modelos lineares em espaço de estados restringida à matriz identidade prendeu-se com questões de tempo e de simplicidade. Desta forma com a matriz C igual à matriz identidade conclui-se que $y(k) = x(k)$ sendo que $y(k)$ é o vector da saída real e $x(k)$ é o vector de estados.

De forma a garantir que os modelos lineares correspondam à realidade, realizou-se a validação dos mesmos. Para este fim retiraram-se novos dados, dados de validação, utilizando o mesmo procedimento anteriormente descrito. A partir destes dados utilizaram-se as acções de controlo obtidas de forma a proceder à excitação dos modelos lineares em espaço de estados com o objectivo de comparar a saída dos modelos com a saída real do processo Amira DTS200 obtida a partir dos dados de validação. Deste modo apresentam-se as validações efectuados nas Figuras 4.6, 4.7 e 4.8:

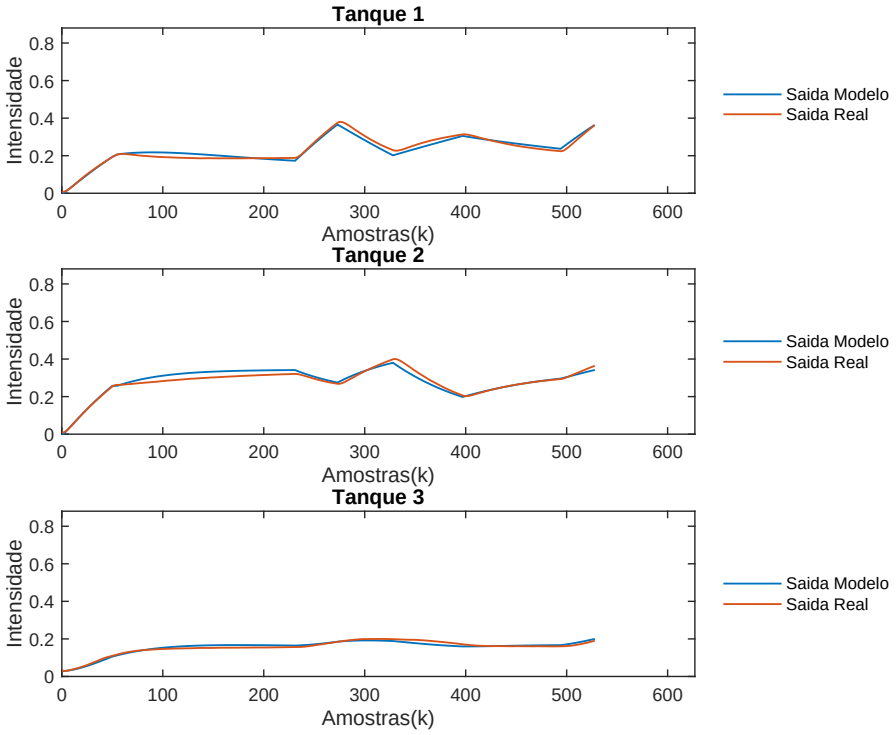

Figura 4.6: Validação do Modelo Linear Zona 1.

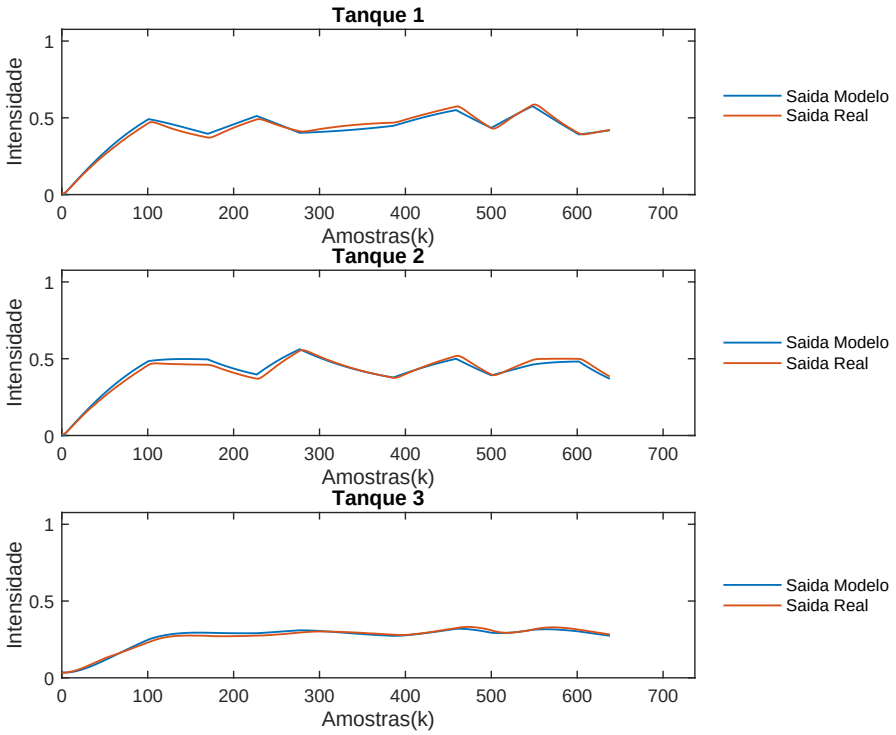

Figura 4.7: Validação do Modelo Linear Zona 2.

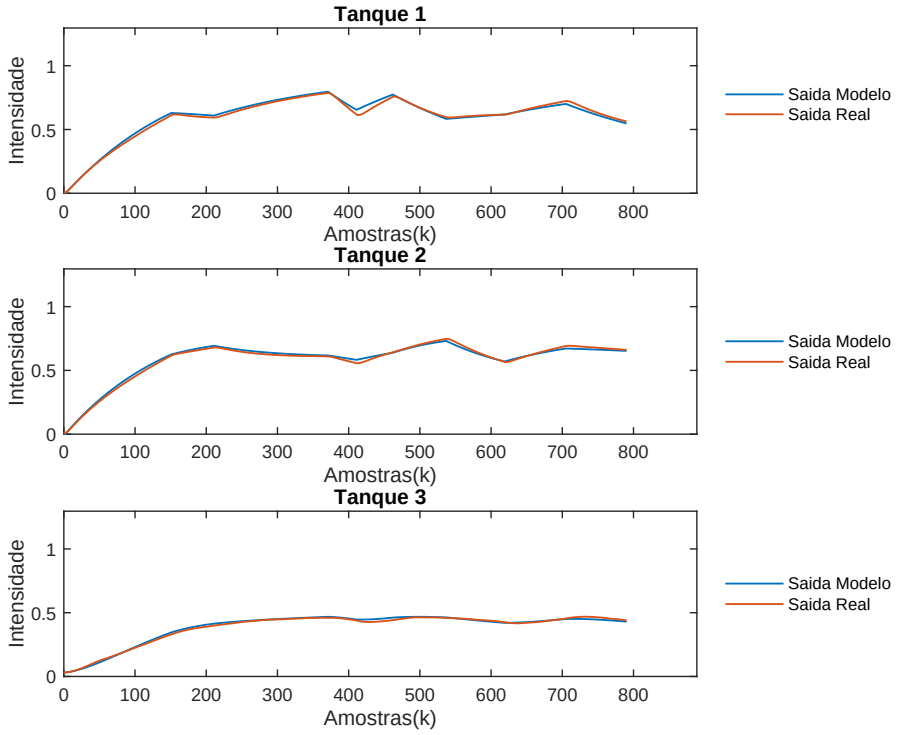

Figura 4.8: Validação do Modelo Linear Zona 3.

Analisando as Figuras 4.6, 4.7 e 4.8 pode-se concluir que os modelos lineares se aproximam bastante da realidade dentro das zonas consideradas. Pode-se constatar que as saídas resultantes dos modelos lineares se aproximam bastante das saídas reais do sistema, tal como é possível verificar na Tabela 4.1 pela ordem de grandeza dos erros obtidos.

Desta forma pode-se concluir pela validação dos modelos lineares que estes são adequados para a utilização no protótipo a construir.

Tabela 4.1: Erros dos modelos lineares nas respectivas zonas

	Zona 1		**Zona 2**		**Zona 3**	
Métricas	**RMSE**	**MSE**	**RMSE**	**MSE**	**RMSE**	**MSE**
Tanque 1	0.0153	2.3268e-04	0.0196	3.8485e-04	0.0172	2.9469e-04
Tanque 2	0.0184	3.3887e-04	0.0210	4.4119e-04	0.0140	1.9605e-04
Tanque 3	0.0096	9.2903e-05	0.0121	1.4572e-04	0.0101	1.0207e-04

4.4 Modelo Não linear

Após ter obtido os modelos lineares construiu-se a arquitectura neuro-difusa, baseada na tipologia ANFIS descrita na Secção 3.7.1, onde os consequentes da inferência difusa são os modelos lineares no espaço de estados onde os antecedentes tomam a forma presente na Figura 4.9:

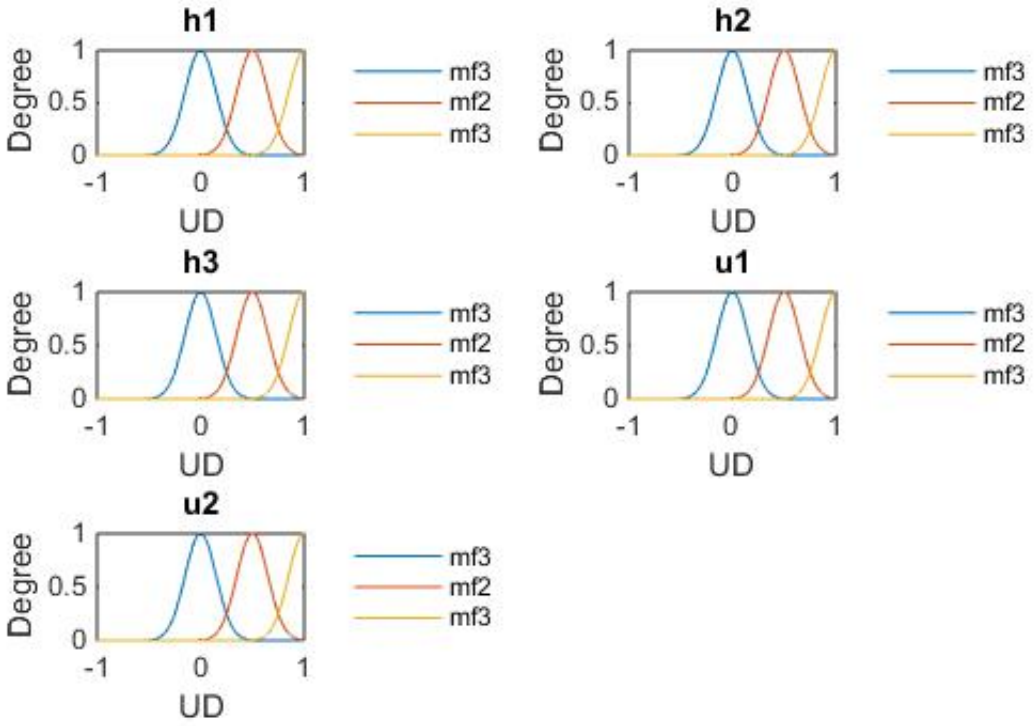

Figura 4.9: Funções de pertença presentes na arquitectura neuro difusa proposta.

Observando a Figura 4.9 pode-se constatar que foram escolhidas três funções de pertença por variável de entrada. Neste caso a arquitectura contém cinco variáveis de entrada sendo estas as duas bombas e os três níveis dos três tanques. Pode-se também constatar que nenhuma função de pertença intersecta mais do que uma função de pertença, e por sua vez esta intersecção encontra-se entre 0,25 e os 0,5 sendo estas regras estritamente necessárias para se construir os antecedentes de uma inferência difusa Cox et al. 1999. Para além disso pode-se constatar que todo o universo de discurso (UD) contém pelo menos uma intersecção com uma função de pertença garantindo assim que haja pelo menos uma regra a disparar. Pode-se ainda concluir que com 5 variáveis de entrada e três funções de pertença por variável de entrada, tem-se 243 regras associadas à inferência difusa presente no modelo neuro difuso.

De forma a adaptar o modelo não linear ao sistema DTS200, treinou-se a rede neuronal adaptando os antecedentes e os consequentes do modelo não linear com base na arquitectura desenvolvida neste trabalho.

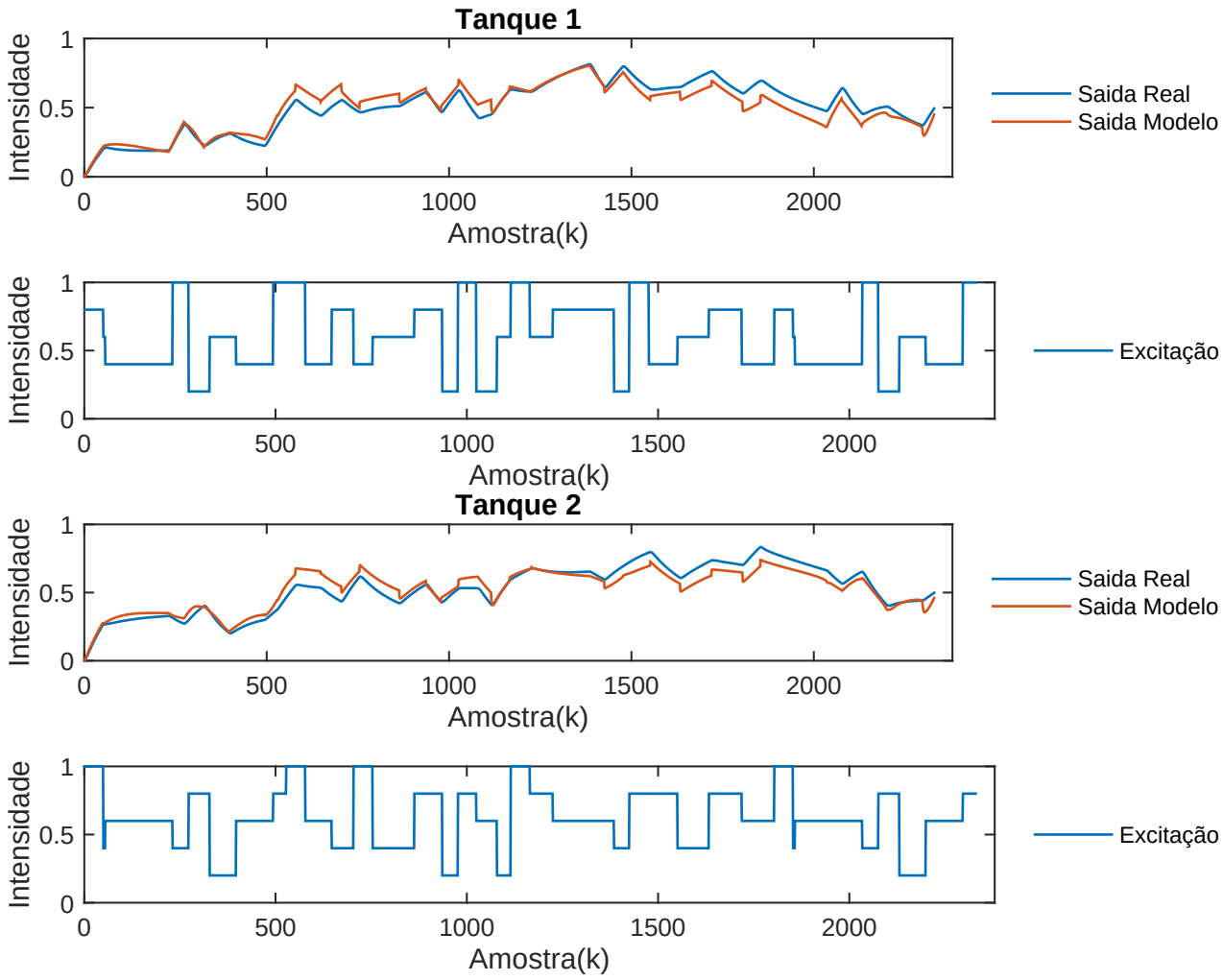

Figura 4.10: Modelo Não Linear: Treino.

De modo a implementar o treino estudou-se alguns algoritmos de treino de sistemas neuro difusos de forma a poder escolher o melhor algoritmo de treino. Uma possibilidade seria a utilização do algoritmo *standard* do ANFIS que utiliza o algoritmo de gradiente descendente com o algoritmo *backpropagation* para optimizar os antecedentes do modelo ou seja proceder à alteração das funções de pertença da inferência difusa e utilizar o algoritmo LSE para os consequentes. No entanto, com base no trabalho desenvolvido por Karaboga e Kaya Karaboga e Kaya 2018 pode-se constatar que o algoritmo standard não se revela o melhor algoritmo devido ao sobreajuste do treino. Assim foi escolhido um algoritmo PSO (Particle Swarm Optimization) em conjunto com o algoritmo *backpropagation* de forma a treinar os antecedentes do protótipo desenvolvido e utilizou-se o PSO para o treino dos consequentes tal como já tinha sido referido na secção 3.7.2.

Para adaptar o modelo neuro difuso ao processo Amira DTS200 retiraram-se os dados de treino excitando o processo em questão. Retiraram-se também os dados de validação

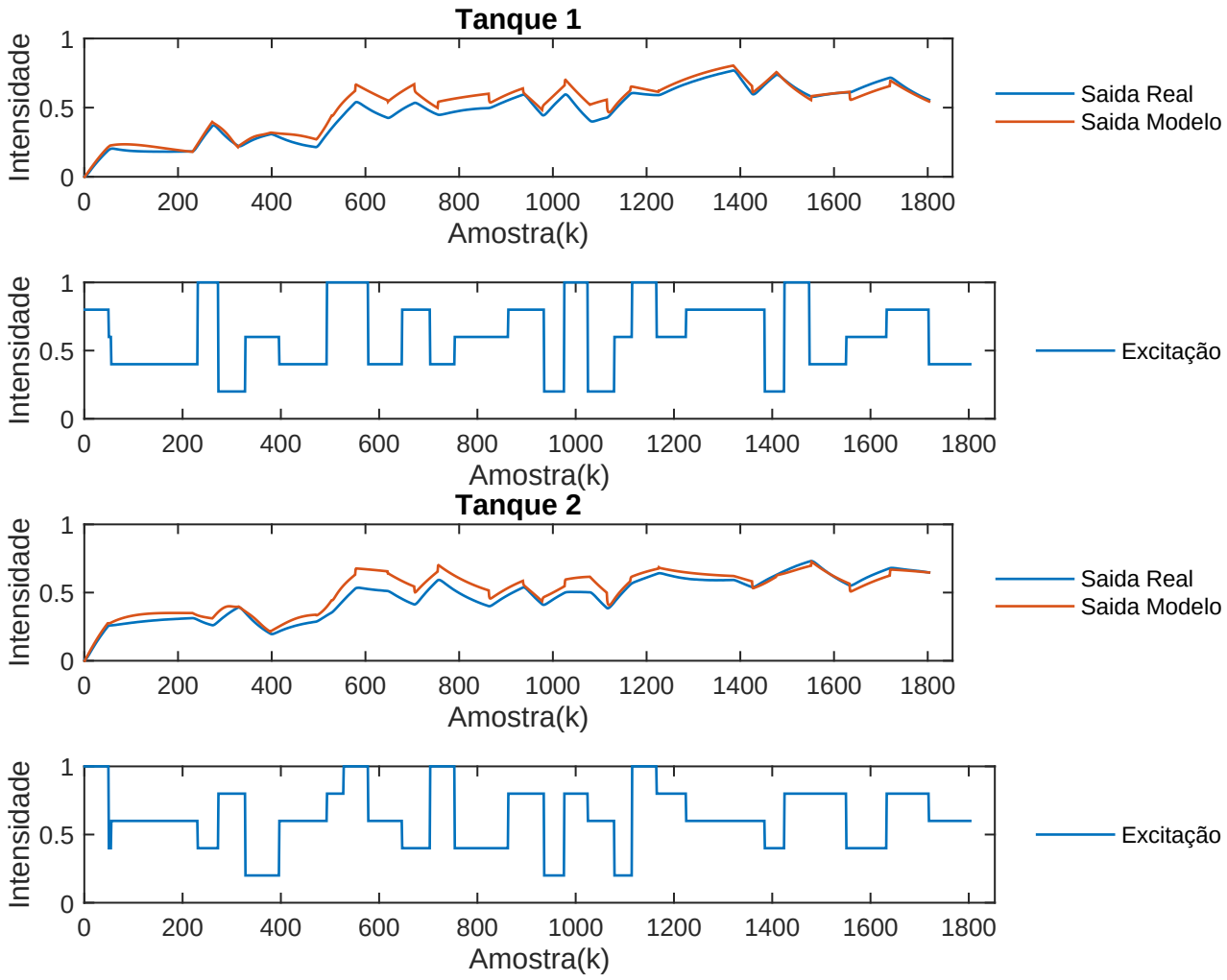

Figura 4.11: Modelo Não Linear: Validação.

de forma a testar se o modelo não linear foi bem treinado. Tal corresponde a verificar se o modelo não linear contém as dinâmicas necessárias para se poder afirmar que o modelo não linear representa adequadamente o sistema Amira DTS200.

Na Figura 4.10 encontra-se representada a resposta do modelo neuro difuso aos dados de treino e na Figura 4.11 os dados correspondentes à resposta do modelo neuro difuso aos dados de validação. Com objectivo de avaliar a adaptação do modelo ao processo foi realizada correctamente apresenta-se também as Tabelas 4.2 e 4.3 com as métricas de desempenho calculadas.

Por análise dos resultados apresentados nas Tabelas 4.2 e 4.3 pode-se constatar que o erro quadrático médio (i.e.*Mean Square Error* (MSE) é menor que a raiz quadrada do erro quadrático médio(i.e.RMSE) quer para o treino do modelo não linear quer para a sua validação. Tal facto seria de esperar uma vez que o RMSE corresponde à raiz quadrada do MSE. A raiz quadrada do erro quadrático médio constitui uma boa métrica a utilizar,

uma vez que torna a escala dos erros da mesma magnitude da escala dos valores alvo. No cálculo dos erros estes são elevados ao quadrado antes de se proceder ao cálculo da média e como tal o RMSE atribui um peso relativamente elevado a erros grandes e como tal constitui uma boa métrica a utilizar quando erros elevados são particularmente indesejados. Ambas as métricas de desempenho permitem verificar que o modelo não linear se aproxima dos valores reais, uma vez que os erros são baixos.

Analisando as Figuras 4.10 e 4.11, pode-se concluir que o modelo neuro difuso foi optimizado satisfatoriamente, apesar deste não ser perfeito. Para se obter um modelo não linear perfeito através de modelos locais lineares, obriga a que o número de modelos lineares a utilizar tenda para infinito. Uma vez que os modelos lineares aproximam o processo a um sistema linear, e uma vez que o processo DTS200 é não linear, só se pode aproximar esta instalação a um modelo linear num ponto de funcionamento. Assim sendo quantos mais pontos de funcionamento forem utilizados, maior será o número de modelos lineares e melhor será a modelação não linear.

A escolha do número de modelos a utilizar está relacionada com a relação benefício/custo uma vez que quanto maior for o número de modelos maior será o tempo e consequentemente o custo da sua implementação. Neste caso particular pode-se concluir que a escolha dos três modelos lineares se manifestou satisfatória com vista a simular o comportamento real da instalação.

Dado que o objectivo é a construção de um controlador neuro difuso utilizando controladores com efeito integral este modelo será considerado aceitável para a finalidade em questão.

Como já foi anteriormente referido vai-se projectar um controlador neuro difuso com base no ANFIS, como foi descrito na Secção 3.7.3, para controlar o processo DTS200, o que faz com que para projectar este controlador seja necessário garantir a estabilidade global do sistema tal como já foi explicado na secção 3.3. Para garantir a estabilidade irá utilizar-se as LMI's, anteriormente referidas na secção 3.4.

Tabela 4.2: Métrica de desempenho do treino do modelo não linear

	MSE	**RMSE**
Tanque 1	0,0042	0,0649
Tanque 2	0,0039	0,0627
Tanque 3	0,0024	0,0488

Tabela 4.3: Métrica de desempenho da validação do modelo não linear

	MSE	**RMSE**
Tanque 1	0,0037	0,0612
Tanque 2	0,0042	0,0650
Tanque 3	0,0028	0,0538

4.5 Controlador Neuro Difuso Proposto

Para a construção do controlador neuro difuso proposto, de forma a garantir a estabilidade global do sistema, utilizou-se a LMI Toolbox do matlab formulando as equações 3.15 e 3.16 de forma a calcular os ganhos do controlador com efeito integral. Devido ao facto das equações garantirem apenas que o controlador é globalmente estável foi necessário restringir o algoritmo ao semi-plano positivo, e dentro de uma região, de forma a que o controlador tenha um desempenho aceitável. Com este fim, na Figura 4.12 apresenta-se o diagrama de lugar de raízes, com as restrições de desempenho.

Através da observação da Figura 4.12, pode-se concluir que a estabilidade do sistema global se verifica uma vez que os pólos do sistema se encontram dentro do círculo unitário. Pode-se também verificar as restrições implementadas a nível do desempenho do controlador. Estas restrições como se pode comprovar pela Figura 4.12 restringem tanto o valor real dos pólos como o valor imaginário dos pólos. Em relação à restrição da parte real, o procedimento foi implementado de forma que os pólos não se encontrassem no limiar da estabilidade, e por sua vez se encontrassem dentro do semi-plano positivo. Em relação às restrições da componente imaginária restringiu-se de forma a que os pólos tivessem o mínimo de componente imaginária possível.

Estas restrições foram implementadas utilizando um algoritmo PSO minimizando o erro de controlo do controlador desenvolvido nesta dissertação utilizando o modelo neuro difuso implementado anteriormente. Pode-se também concluir que estas restrições foram implementadas de forma a garantir que o controlador não teria oscilações na sua resposta.

Após verificar a estabilidade global do sistema, e após alguns testes para verificar o desempenho do controlador através do algoritmo PSO e da implementação das restrições no algoritmo das LMI's, vai-se implementar o anel de controlo.

O anel de controlo pode ser implementado em simulação e em modo *online*, sendo que a principal diferença entre eles, consiste na obtenção da saída. No caso do anel de controlo ser implementado para o modo de funcionamento em simulação, utiliza-se o modelo neuro difuso proposto para emular o sistema Amira DTS200 calculando a saída do sistema. Já para o modo de funcionamento *online* utiliza-se o processo real Amira DTS200 para obter a saída do sistema. Após obtenção da saída do sistema, quer seja em modo de simulação ou *online*, esta é fornecida ao controlador neuro difuso proposto de forma a este poder calcular as acções de controlo a aplicar ao modelo neuro difuso no caso simulado, e a aplicar nas bombas do processo Amira DTS200 no caso *online*.

A principal razão pela qual se construiu o anel de controlo simulado está relacionada com questões de segurança uma vez que os ganhos de controlo calculados não tinham sido validados. Outra razão prende-se com o facto de poder testar o controlador sem necessitar do processo real.

A implementação do anel de controlo em simulação foi assim utilizada como base de partida para se passar ao processo real com utilização do anel de controlo online sendo

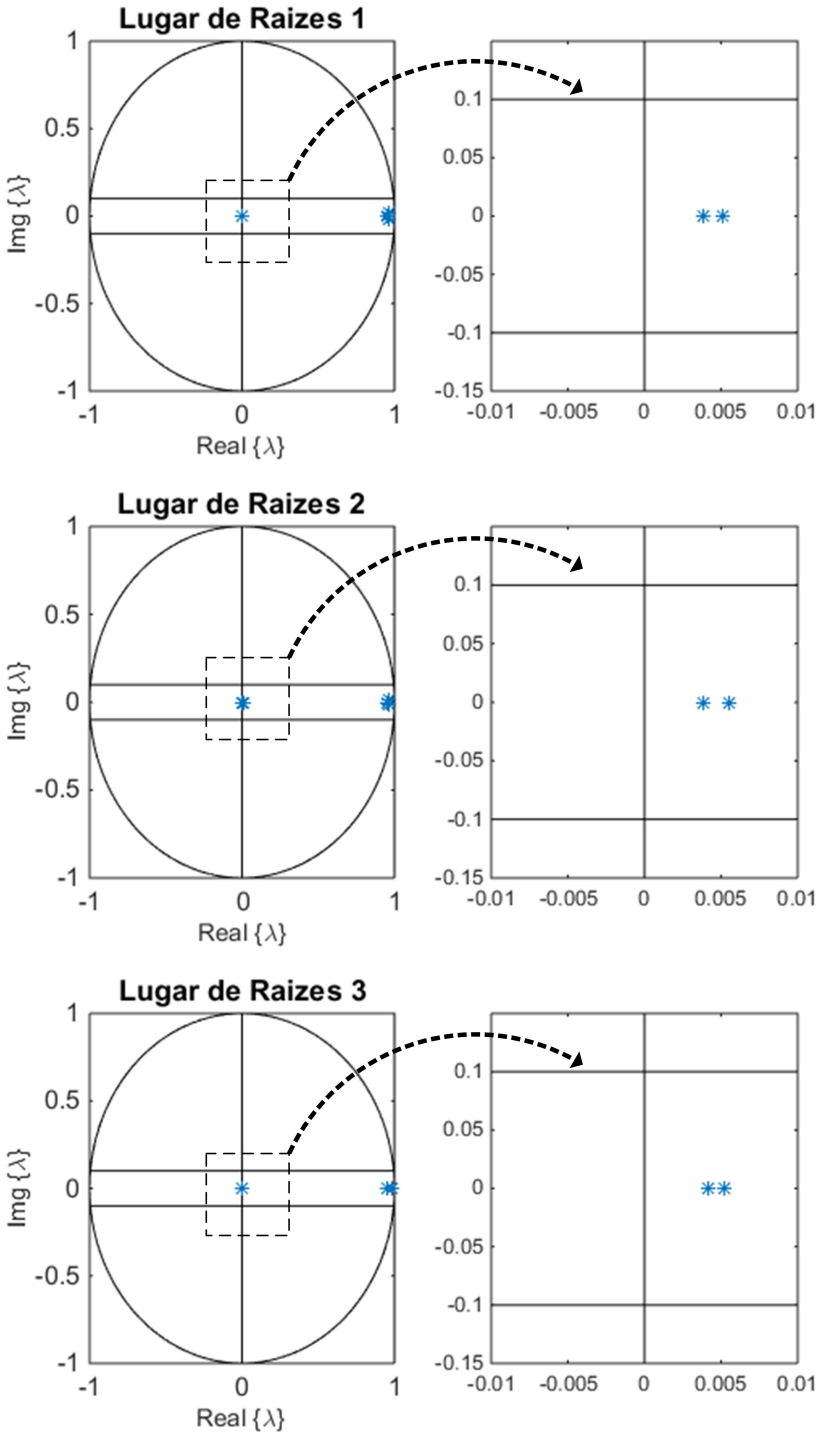

Figura 4.12: Diagrama de Lugar de Raízes com as Restrições de Desempenho

este o principal objectivo.

Na secção seguinte vai-se apresentar a simulação do controlador neuro difuso.

4.5.1 Simulação do Controlador Neuro Difuso

De forma a validar o desempenho do controlador projectado realizou-se uma simulação, em que a saída do sistema é calculada pelo modelo neuro difuso desenvolvido nesta dissertação referido na secção 3.7. O objectivo deste teste consiste em verificar se os ganhos do sistema foram bem calculados.

Apresenta-se na figura 4.13 e na tabela 4.4 os resultados da simulação:

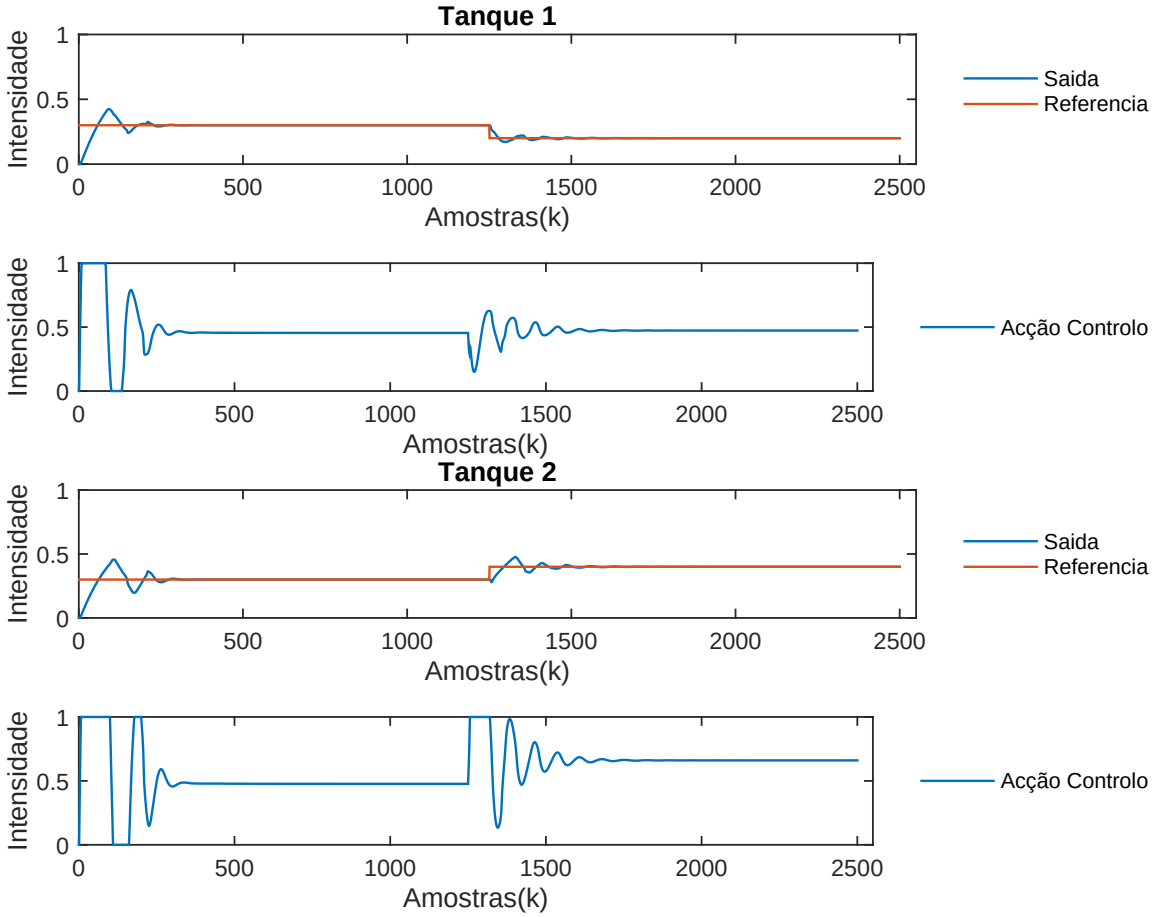

Figura 4.13: Simulação do Controlador Neuro Difuso Proposto utilizando o modelo neuro difuso.

Analisando a Figura 4.13 e a Tabela 4.4, pode-se concluir que o controlador foi sintonizado com sucesso, uma vez que quer a saída do tanque 1, quer a saída do tanque 2 convergem para a referência estabelecida, tendo um erro em regime permanente igual a zero. Este facto deve-se à implementação do efeito integral nos controladores de retroacção de variáveis de estado. Verifica-se também que o erro da saída em relação às

Tabela 4.4: Métricas de desempenho do controlador neuro difuso em simulação

Métricas	Tanque 1	Tanque 2
RMSE	0,0317	0,0368
MSI	0,0023	0,0033
Energia Mínima	0,2413	0,3802
Sobreelevação [%]	41,523	52,094
Tempo de establecimento [s]	255,426	292,470
Tempo de Subida [s]	41,361	44,490

referências encontra-se inferior a 5 %, o que permite concluir que os ganhos de controlo foram bem parametrizados.

Em relação à variação da acção de controlo e ao parâmetro da energia mínima pode-se concluir que estes valores são baixos o que indica que este controlador pode ser uma solução viável, pois apresenta uma boa eficiência energética.

Analisando com mais detalhe em relação às métricas de desempenho de controladores pode-se concluir que este controlador apresenta uma sobre-elevação elevada de cerca de 50 % e por sua vez um tempo de estabelecimento superior a 260 segundos. Este facto deve-se à acumulação do erro o que leva a uma saturação da acção de controlo. Esta acumulação pode-se resolver com um sistema de *anti-windup* reduzindo a acumulação do erro. Em relação ao tempo de subida do sistema este ronda os 45 segundos o que é bastante aceitável para o processo Amira DTS200.

Na secção seguinte será apresentada a simulação do controlador neuro difuso proposto em modo online.

4.5.2 Controlador Neuro Difuso Proposto em linha

Após validar o controlador em modo offline, foi-se testar o controlador em modo online no processo Amira DTS200. Para tal foram utilizados os sensores de nível de cada tanque para medir o nível de água, o modelo neuro difuso para ponderar as acções de controlo calculadas, e utilizou-se ainda as duas bombas para actuar sobre o sistema físico.

Assim sendo na Figura 4.14 e na Tabela 4.5 apresentam-se os resultados obtidos no teste.

Tabela 4.5: Métrica de desempenho do controlador neuro difuso no processo

Métricas	Tanque 1	Tanque 2
RMSE	0,0511	0,0531
MSI	0,0051	0,0054
Energia Mínima	0,4490	0,2932
Sobreelevação [%]	43,221	50,470
Tempo de establecimento [s]	246,520	266,357
Tempo de Subida [s]	46,892	47,377

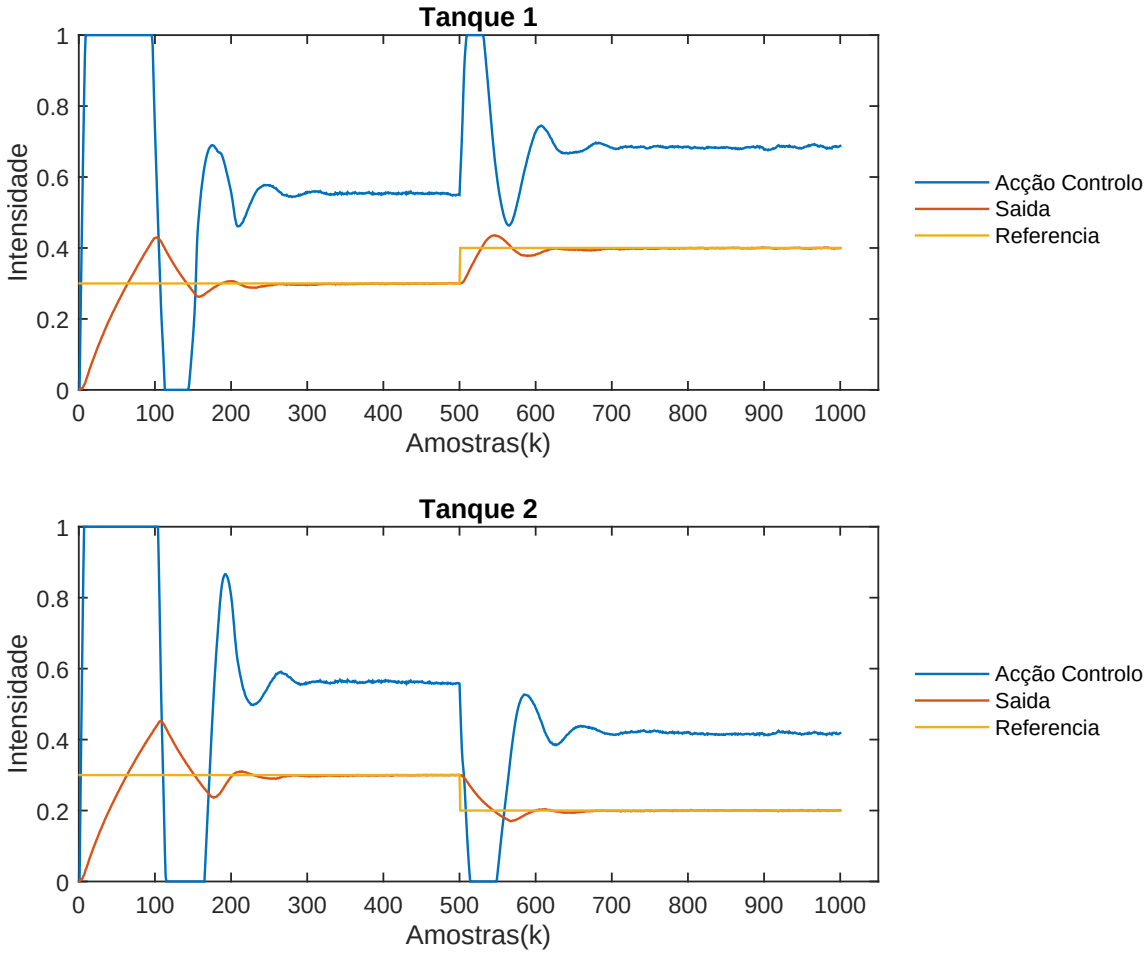

Figura 4.14: Teste do Controlador Neuro Difuso no processo.

Observando a Figura 4.14 e a Tabela de desempenho 4.5, pode-se verificar que o controlador proposto apresenta resultados satisfatórios pois a saída do tanque 1 e do tanque 2 convergem para a referência estabelecida.

Em relação ao cálculo da raiz quadrada do erro quadrático médio pode-se concluir que este é sempre inferior a 6% quer para o tanque 1 quer para o tanque 2, o que nos leva a constatar que para o sistema Amira DTS200 este controlador é uma solução bastante viável, uma vez que baixos valores desta métrica significam menor diferença entre a saída e a referência estabelecida.

Pode-se também comparar a discrepância de resultados entre o teste do controlador no processo e a simulação do controlador utilizando o modelo neuro difuso. As diferenças observadas devem-se ao facto dos modelos lineares considerarem as duas bombas lineares e não contarem com a zona morta das bombas o que faz com que o *overshoot* do controlador em simulação seja significativamente mais alto do que o teste do controlador no processo. Ainda tendo em consideração o mesmo facto faz com que o tempo de estabelecimento do teste do controlador no processo seja muito maior que a simulação do

controlador com o modelo neuro difuso pois o modelo acaba por ter mais poder de actuação das bombas o que faz com que o controlador em simulação tenha um comportamento mais rápido, o que na realidade não acontece. Em suma esta discrepância entre as duas simulações deve-se ao facto de erros de modelação.

Conclui-se também que o controlador projectado apresenta um valor da sobreelevação e um tempo de estabelecimento bastante elevado. Este facto deve-se ao facto do controlador neuro difuso ser construído através de controladores por retroacção de variáveis de estado com efeito integral.

Os controladores com efeito integral muitas vezes fazem com que a sobre elevação seja elevada e consequentemente o tempo de estabelecimento aumente, pois o integral do erro acumulado vai ser muito grande, o que faz com que a acção de controlo sature originando um tempo para descarregar o erro acumulado, o que leva ao aumento da sobreelevação. Assim sendo e de forma a reduzir o valor da sobreelevação houve a necessidade de implementar um sistema de *anti-windup*.

4.5.3 Controlador Neuro Difuso com *Anti-Windup*

Como foi anteriormente referido, de forma a reduzir a sobreelevação do controlador decidiu-se implementar um sistema com *anti-windup*.

Na literatura pode-se encontrar diversas arquitecturas para implementar um sistema deste tipo Ohr 2003. Com este objectivo a implementação escolhida foi a saturação do ganho integral consoante o valor do erro acumulado. Assim sendo implementou-se o sistema *anti-windup* com base na equação 4.7 para o cálculo da componente integral da acção de controlo:

$$u_{integral}(k) = \varepsilon * e^{-(|\varepsilon * \alpha|)} * K_i \tag{4.7}$$

Analisando a equação 4.7, onde $\varepsilon = \sum_1^k r(k) - y(k)$, corresponde ao erro acumulado, $r(k)$ e $y(k)$ correspondem à referência e saída do sistema respectivamente, K_i ao ganho da componente integral e α é uma constante que faz com que mude o valor da restrição de K_i em relação ao valor do erro acumulado.

De forma a compreender melhor a restrição apresenta-se a figura 4.15:

Analisando a Figura 4.15 pode-se constatar que quanto maior for o valor de α mais restritiva é a acumulação do erro, e por sua vez menor será a componente integral na acção de controlo a aplicar ao sistema.

A selecção do valor de α depende do número de acções de controlo integrais presentes na arquitectura de controlo. Devido ao facto da arquitectura proposta ter sido construida para o sistema Amira DTS200, leva à existência de duas acções de controlo integrais. O facto de haver duas acções de controlo integrais advém da existência de dois actuadores (i.e.duas bombas), presentes no sistema Amira DTS200. Assim sendo, houve a necessidade de sintonizar dois valores de $alpha$ um para cada bomba, $alpha_1$ que corresponde à bomba

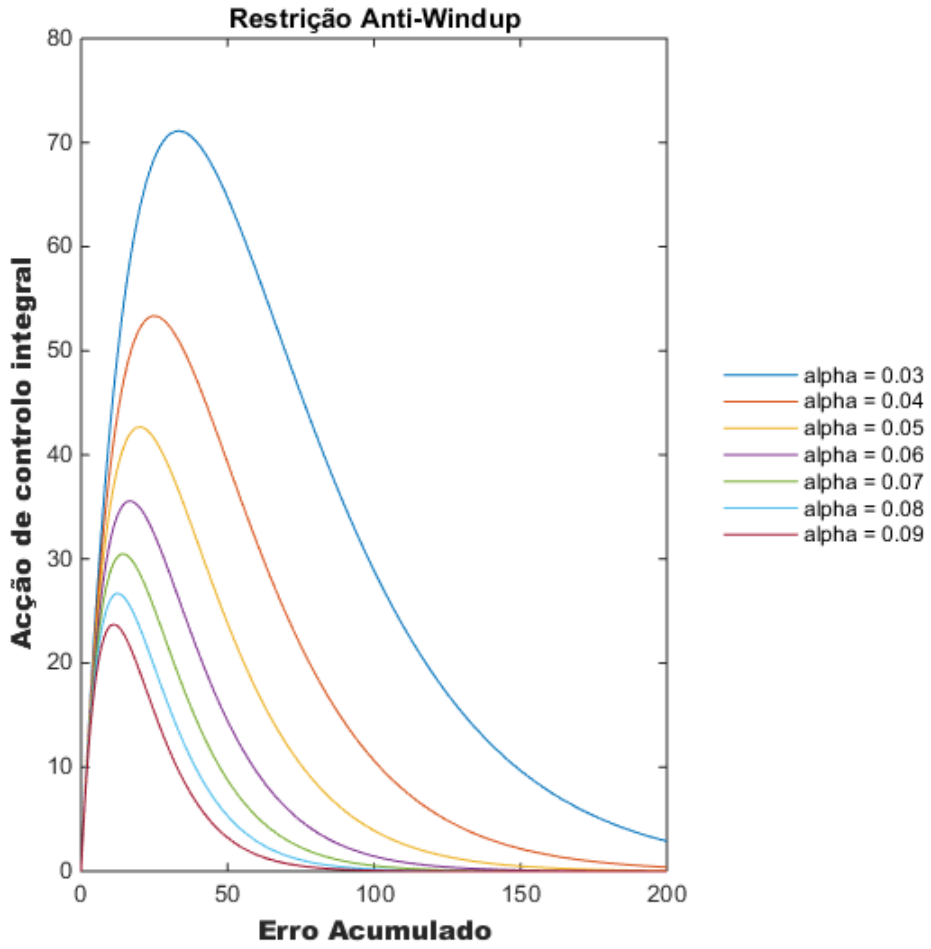

Figura 4.15: Influência do α na restrição da acção de controlo integral.

do tanque 1 e $alpha_2$ que corresponde à bomba do tanque 2. Tal necessidade deve-se ao facto das duas bombas apresentarem dinâmicas distintas.

Analisando a equação 4.7 e verificando a forma como α influencia a acumulação do erro, realizaram-se vários testes de forma a calcular os valores de α que garantissem uma sobreelevação perto dos 5 % e a minimização do erro de controlo. Com vista a alcançar esse objectivo sintonizaram-se os valores de α com um algoritmo PSO, com a função de custo do algoritmo definida como o menor erro da saída em relação à referência, e de forma que a sobreelevação tendesse para os 5 %.

De seguida apresenta-se na Figura 4.16 a simulação do controlador com o sistema *anti-windup* implementado e na Tabela 4.6 as métricas de desempenho do controlador.

Analisando a Figura 4.16 e a Tabela 4.6 pode-se concluir que a implementação do sistema *anti-windup* foi bem sucedida melhorando o desempenho do controlador, mantendo o erro em regime permanente nulo, diminuindo significativamente a sobreelevação dos dois tanques, ficando esta próxima dos 10 %, e o tempo de estabelecimento em cerca de 146 segundos. Conseguiu-se também reduzir a raiz quadrada do erro quadrático médio

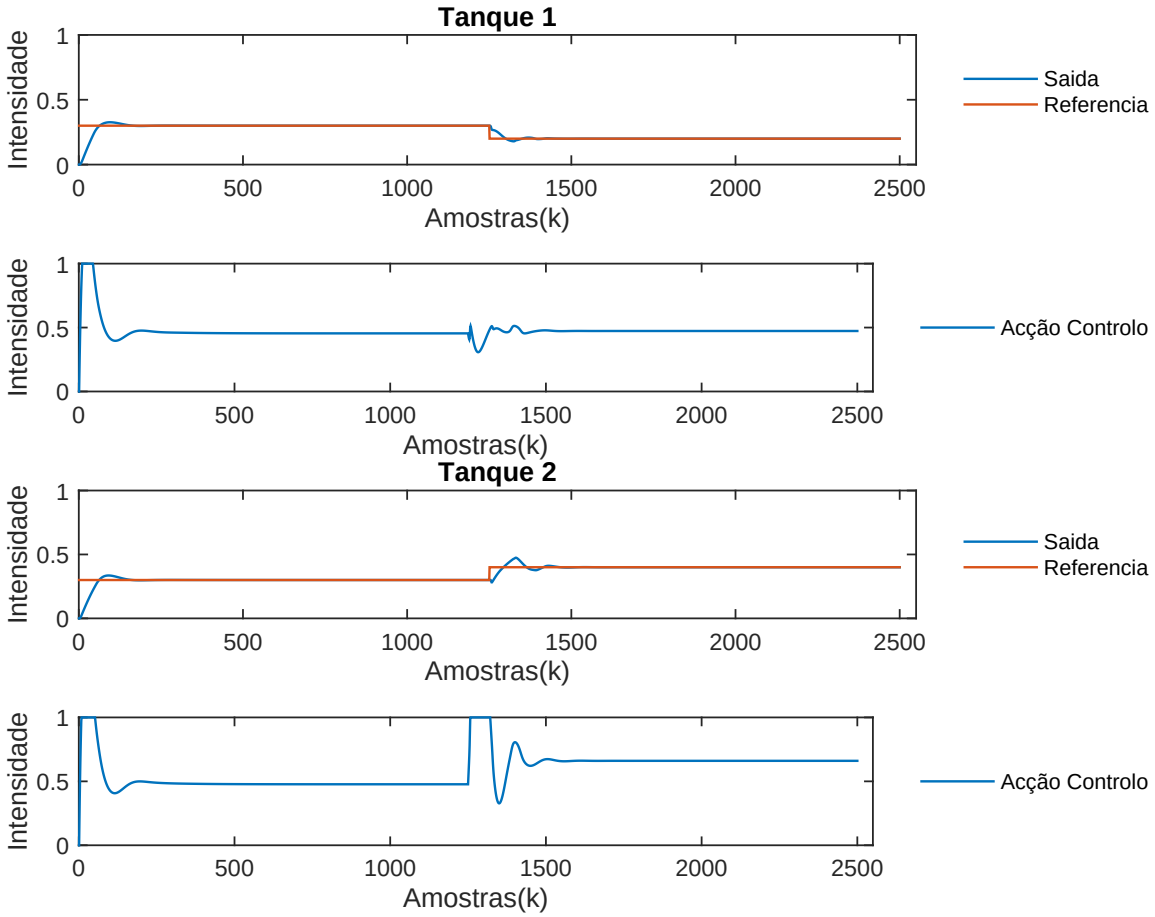

Figura 4.16: Simulação do Controlador Neuro Difuso Proposto com $\alpha 1 = 0,07$ e $\alpha_2 = 0,09$.

Tabela 4.6: Métrica de desempenho do controlador neuro difuso proposto com $\alpha 1 = 0,07$ e $\alpha_2 = 0,09$.

Métricas	Tanque 1	Tanque 2
RMSE	0,0290	0,0304
MSI	9,8702e-04	0,0015
Energia Mínima	0,2299	0,3625
Sobreelevação [%]	8,7424	11,7109
Tempo de estabelecimento [s]	145,8133	145,0926
Tempo de Subida [s]	42,3226	44,4041

correspondente à diferença entre a saída e a referência, e por sua vez minimizar a variação da acção de controlo bem como a energia mínima.

Em suma conclui-se que a introdução deste sistema de *anti-windup* veio proporcionar uma melhoria em todas as métricas de desempenho seleccionadas.

Após se verificar que a implementação do sistema *anti-windup* foi conseguida com sucesso testou-se o controlador neuro difuso proposto com o sistema de *anti-windup* no

sistema AMIRA DTS200.

Assim sendo na Figura 4.17 e na Tabela 4.7 apresentam-se os resultados desta simulação:

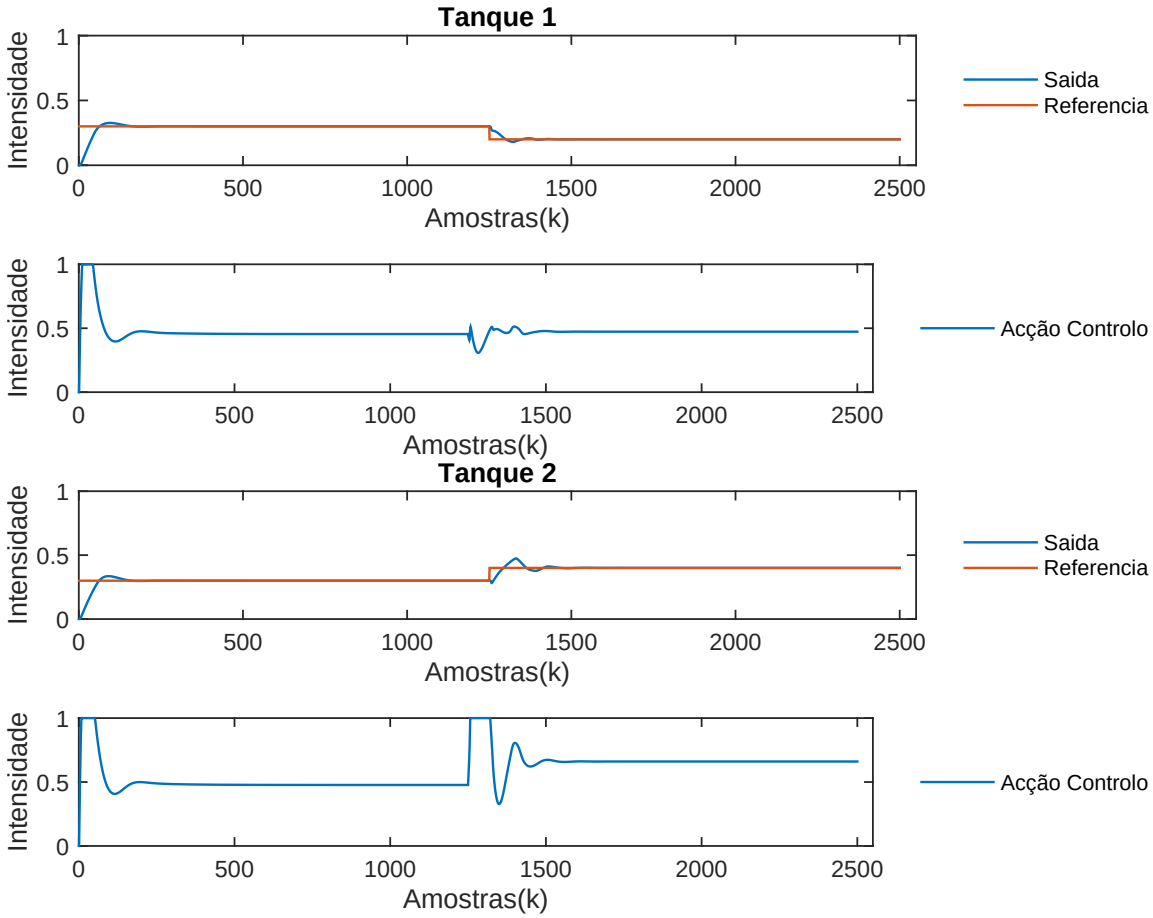

Figura 4.17: Teste no processo do Controlador Neuro Difuso Proposto com $\alpha 1 = 0,07$ e $\alpha_2 = 0,09$.

Tabela 4.7: Métrica de desempenho do controlador neuro difuso proposto no processo com $\alpha 1 = 0,07$ e $\alpha_2 = 0,09$.

Métricas	**Tanque 1**	**Tanque 2**
RMSE	0,0297	0,0301
MSI	0,0018	0,0025
Energia Mínima	0,2262	0,3889
Sobreelevação [%]	8,7362	12,7894
Tempo de estabelecimento [s]	112,6140	152,0173
Tempo de Subida [s]	42,8371	44,3128

Observando a Figura 4.17 e a Tabela, 4.7 pode-se concluir que o sistema de *anti-windup* veio proporcionar melhorias significativas ao controlador neuro difuso proposto. Tal é facilmente observável por comparação com a Tabela, 4.5. Por análise comparativa das tabelas pode-se concluir que a adição do sistema *anti-windup* permitiu reduzir a sobrelevação do sistema com *anti-windup* em cerca de 5 vezes quando comparado com o sistema sem *anti-windup*. Permitiu também uma diminuição considerável do tempo de estabelecimento.

A redução da sobreelevação e do tempo de estabelecimento levou á redução da raiz quadrada do erro quadrático médio e também a uma redução significativa em relação à variação da acção de controlo assim como uma redução da energia mínima.

Em suma a adição do sistema *anti-windup* revelou-se uma mais valia no protótipo desenvolvido nesta dissertação devido ao facto do sistema *anti-windup* proposto neste trabalho reduzir o erro de controlo, reduzir a variação de controlo e ainda reduzir a energia mínima. Com esta abordagem foi possível tornar o controlador mais eficiente a todos os níveis.

Apesar deste controlador neuro difuso desenvolvido neste trabalho ser considerado eficiente, seria conveniente a sua comparação com outros tipos de controladores já desenvolvidos.

Neste sentido, na secção seguinte, apresenta-se os resultados de simulação de um controlador PID Difuso adaptado à configuração escolhida nesta dissertação, com o qual o controlador desenvolvido vai ser comparado.

4.6 Controlador PID Difuso em Linha

Como já foi anteriormente referido, de forma avaliar o desempenho do controlador neuro difuso projectado vai-se comparar com um controlador proposto em *Aplicação de técnicas de controlo óptimo difuso em ambientes distribuídos*. Este por sua vez é um controlador proporcional integral derivativo difuso, e teve de ser adaptado, pois este foi sintonizado para outra configuração. Após ter adaptado o controlador PID difuso apresenta-se na figura 4.18 e na tabela 4.8 o desempenho deste controlador adaptado à configuração escolhida nesta dissertação:

Tabela 4.8: Métrica desempenho do controlador PID Difuso no processo

Métricas	**Tanque 1**	**Tanque 2**
RMSE	0,0614	0,0666
MSI	0,2254	0,3862
Energia Mínima	0,2262	0,3889
Sobreelevação [%]	0,8962	0,1691
Tempo de establecimento [s]	62,0115	68,1706
Tempo de Subida [s]	44,0258	47,0081

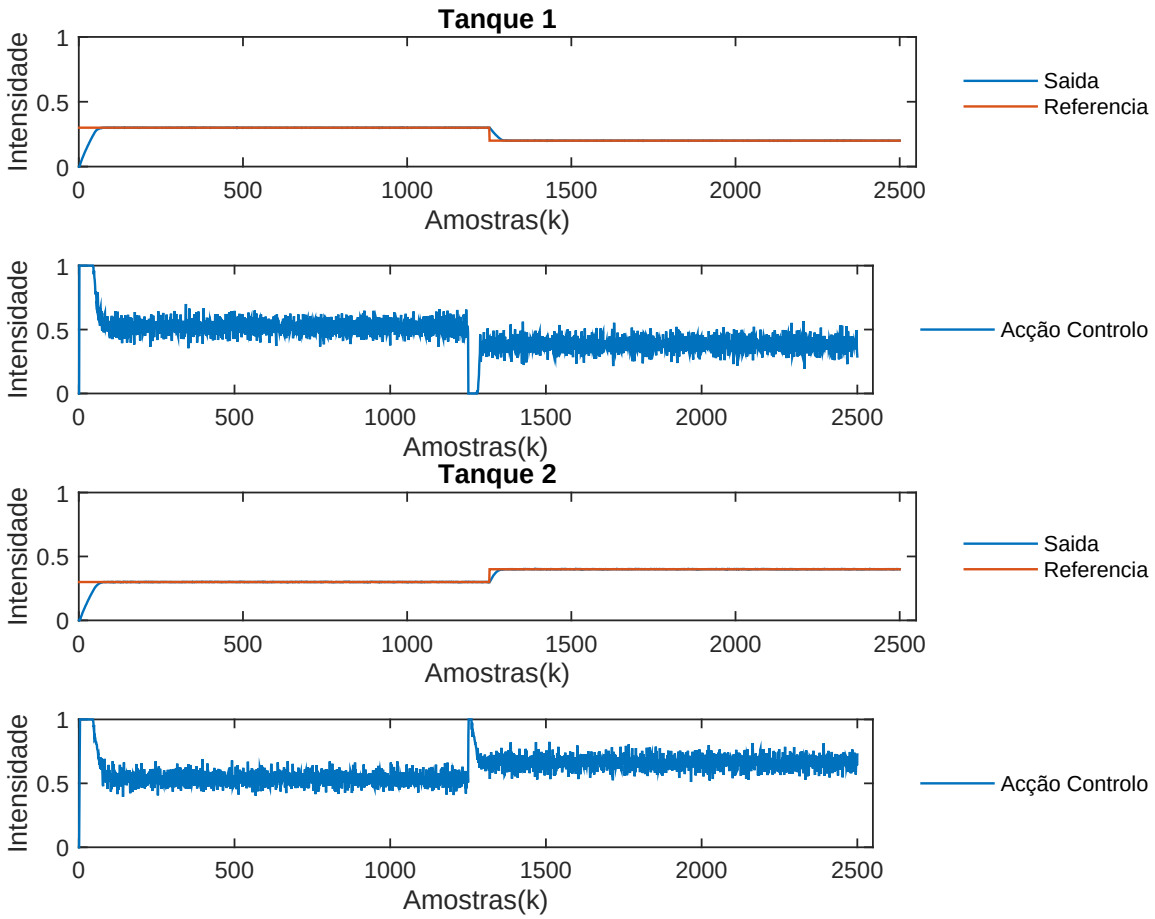

Figura 4.18: Teste do Controlador PID Difuso no processo

Analisando a figura 4.18 pode-se constatar que o controlador PID difuso foi bem sintonizado para a configuração escolhida nesta dissertação, pois as saídas dos tanque 1 e do tanque 2 tendem para as referências de controlo, sendo que o erro em regime permanente é nulo. Pode-se analisar também que este controlador a nível da acção de controlo das duas bombas apresenta uma grande variação o que leva a um gasto de energia e um desgaste das bombas mais acentuado. Observando a tabela 4.8 é possível concluir que o erros calculados das saídas em relação às referências são inferiores a 7 % o que permite concluir que este controlador é satisfatório. Em relação à variação da acção de controlo e ao cálculo da energia mínima o mesmo não se pode concluir pois este apresenta valores na ordem dos 0,23 para o tanque 1, 0,39 para o tanque 2 e 0,27 para o tanque 1 e 0,39 para o tanque 2 respectivamente. Em relação às especificações de controlo este controlador apresenta um tempo de estabelecimento e uma sobre-elevação bastante satisfatórios na ordem dos 62 segundos para o tanque 1, 68 segundos para o tanque 2 e 0,90 % para o tanque 1, 0,17 % para o tanque 2 respectivamente.

Assim sendo na secção seguinte vai-se comparar todos os controladores implementados nesta dissertação.

Na secção seguinte faz-se uma comparação de todos os controladores implementados e testados no processo Amira DTS200 utilizando as métricas de desempenho.

4.7 Comparação Controlador Neuro Difuso Proposto vs. PID Difuso

Após ter implementado e validado os controladores neuro difuso proposto, controlador neuro difuso proposto com *anti-windup* e o PID Difuso vai-se nesta secção comparar os resultados dos mesmos.

Com este objectivo apresenta-se as tabelas 4.9 e 4.10 com o desempenho de cada controlador.

Tabela 4.9: Métrica desempenho dos controladores Neuro Difuso Proposto VS PID Difuso

Tanque 1			
Controlador	**ANFIS**	**ANFIS anti-windup**	**PID Difuso**
RMSE	0,0511	0,0297	0,0614
Energia Minima	0,4490	0,2262	0,2262
Variação Acção Controlo	0,0051	0,0018	0,2254
Sobreelevação	43,221	8,7362	0,8962
Tempo de establecimento	246,520	112,614	62,0115
Tempo de Subida	46,892	42,8371	44,0258

Tabela 4.10: Métrica desempenho dos controladores Neuro Difuso Proposto VS PID Difuso

Tanque 2			
Controlador	**Proposto**	**Proposto *anti-windup***	**PID Difuso**
RMSE	0,0531	0,0301	0,0666
Energia Minima	0,2932	0,3889	0,3889
MSI	0,0054	0,0025	0,3862
Sobreelevação	50,470	12,7894	0,1691
Tempo de establecimento	266,357	152,0173	68,1706
Tempo de Subida	47,377	44,3128	47,0081

Analisando as tabelas 4.9 e 4.10 pode-se constatar que o controlador neuro difuso proposto sem implementação de *anti-windup* apresenta os piores valores em relação a todas a métricas de desempenho com excepção da métrica da variação da acção de controlo comparando com o PID Difuso. Assim sendo a nível de especificação o pior controlador acaba por ser o controlador neuro difuso proposto sem o sistema *anti-windup*. Esta conclusão deve-se ao facto deste saturar a ação de controlo, e saturar a acumulação do erro da saída em relação à referência.

Em relação ao controlador neuro difuso proposto com *anti-windup*, este acaba por ser o controlador com o menor erro cerca de 4 % a menos que o controlador PID Difuso. Já em relação à variação da acção de controle verifica-se que o controlador neuro difuso proposto com sistema *anti-windup* apresenta um valor inferior em cerca de 0,38 quando comparado com o controlador PID Difuso o que permite concluir que este controlador terá um desgaste menor em relação às bombas que actuam na instalação.

Em relação a eficiência energética todos os controladores apresentam valores similares o que faz com que esta métrica não seja um factor diferenciador entre estes controladores a projectar.

Já em relação à sobreelevação e ao tempo de estabelecimento o controlador PID Difuso apresenta melhor resultados pois apresenta um tempo de estabelecimento bastante inferior em relação aos outros controladores implementados cerca de 50 segundos mais rápido no tanque 1 e 84 segundos mais rápido no tanque 2 do que o controlador neuro difuso proposto com *anti-windup*. Em suma pode-se concluir que o controlador PID Difuso é um controlador mais rápido a atingir a referência, mas é um controlador que impõe mais variações de acção de controlo provocando um maior desgaste nas bombas da instalação e apresenta um erro superior ao controlador neuro difuso proposto com *anti-windup*.

Comparando os resultados das métricas do tanque 1 com as do tanque 2, teoricamente estes deveriam ser os mesmos, mas tal não acontece devido ao facto das bombas terem tempos de resposta diferentes. Pode-se também concluir que os resultados das métricas calculadas para o tanque 2 apresentam piores valores. Esta conclusão depreende-se do facto do tempo morto da bomba 2 ser significativamente maior que o da bomba 1 o que leva esta discrepância de resultados.

Assim sendo pode-se concluir que o controlador implementado nesta dissertação, apesar de ser mais lento no tempo de resposta que o controlador PID Difuso com inferência do tipo de Mandami com o qual foi comparado, apresenta num entanto algumas vantagens competitivas que o tornam numa boa solução. O controlador neuro difuso proposto com o sistema *anti-windup* que teve como base para a sua implementação a arquitectura neuro difusa ANFIS apresenta um erro inferior quando comparado com os restantes controladores, tal como se observa nas tabelas 4.9 e 4.10. Também a variação da acção de controlo se apresenta inferior em relação aos demais o que torna este controlador uma boa opção de escolha pois também provoca menos desgaste no equipamento sendo este um factor de grande importância em sistemas industriais.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusões

Esta dissertação teve como principal objectivo o desenvolvimento de um protótipo de controlador neuro difuso para dar resposta ao problema de controlo de seguimento para sistemas não lineares.

O protótipo desenvolvido desde a fase de modelação até à fase de construção do controlador neuro difuso foi implementado e foi validado recorrendo à utilização da instalação Amira DTS200 a qual é constituída por um sistema de três tanques.

Para tal foi em primeiro lugar realizada uma revisão bibliográfica sobre as questões de modelação com modelos neuro difusos e projecto de controladores neuro difusos.

Os modelos lineares foram obtidos por identificação em diferido e em diferentes pontos de funcionamento específicos e foram validados para a construção do modelo não linear e consequente treino. Neste trabalho utilizaram-se três zonas especificas de funcionamento de forma a construir os modelos lineares no espaço de estados. Os modelos lineares obtidos foram posteriormente validados de forma a garantir que se apresentam adequados e para tal foram utilizadas as excitações obtidas para proceder à excitação dos modelos lineares, e desta forma foi possível comparar a saída dos modelos com a saída real do processo Amira DTS200 obtida a partir dos dados de validação.

O modelo não linear foi posteriormente desenvolvido e com base nos modelos lineares foi construida uma arquitectura neuro difusa tendo por base a arquitectura ANFIS, sendo os consequentes da inferência difusa os modelos lineares em espaço de estados que foram desenvolvidos e previamente validados, e os antecedentes, as funções de pertença escolhidas para arquitectura neuro difusa proposta. O modelo não linear desenvolvido foi adaptado ao sistema Amira DTS200 treinando para esse fim a rede neuronal tendo sido

para tal utilizado o algoritmo PSO para treino dos consequentes em conjunto com o algoritmo *backpropagation* para treino dos antecedentes. Após treino e validação verificou-se que o modelo não linear representa de forma adequada o processo Amira DTS200.

Apesar do modelo neuro difuso não ser perfeito ele mostrou-se contudo satisfatório para a construção do controlador neuro difuso utilizando efeito integral. Foram determinadas as métricas de desempenho tais como RMSE, MSI, e energia mínima, concluindo-se que as saídas dos tanques convergem para as referências estabelecidas e os valores do RMSE quer para o tanque 1 quer para o tanque 2 não atingem os 6% o que mostra que o controlador pode ser considerado como uma solução muito viável. Contudo uma vez que o controlador apresenta um valor da sobreelevação elevado (i.e. à roda de 50 %) bem como um tempo de estabelecimento elevado (i.e. à roda 250 segundos) fruto da sua construção através de controlador por retroacção de variáveis de estado com efeito integral. Surgiu a sim a necessidade de implementar um sistema de *anti-windup*.

O controlador neuro difuso proposto com o sistema de *anti-windup* foi posteriormente testado no sistema Amira DTS200 de onde foi possível concluir que a adição do sistema *anti-windup* proporcionou uma redução cerca de 5 vezes na sobreelevação quando comparado com o sistema *anti-windup*. Tal conduziu também a uma redução considerável no tempo de estabelecimento. Com a introdução do sistema *anti-windup* foi possível tornar o controlador neuro difuso mais eficiente a vários níveis.

Finalmente o novo controlador neuro difuso desenvolvido foi comparado com um controlador PID Difuso adaptado á configuração seleccionada nesta dissertação para o sistema AmiraDTS200, com o objectivo de avaliar o desempenho do controlador neuro difuso proposto. Verifica-se que o controlador neuro difuso proposto com o sistema de *anti-windup* apresenta um RMSE inferior ao do controlador PID Difuso assim como uma variação da ação de controlo muito inferior com valores da ordem de grandeza 100 vezes inferiores quando comparados com o PID Difuso. Os valores menores da acção de controlo são mais favoráveis ao funcionamento da instalação uma vez que provocaram menor desgaste no *hardware*.

Assim conclui-se que o protótipo desenvolvido nesta dissertação apresenta-se como uma solução bastante viável.

5.2 Trabalho Futuro

Existe ainda um grande trabalho a desenvolver nesta área como a implementação de controladores neuro difusos Robustos, e a integração destes em sistemas comutados.

Os sistemas *Hybrid Dynamical Systems* (HDS) são sistemas dinâmicos que envolvem a iteração de dinâmicas discretas com dinâmicas contínuas permitindo assim representar com precisão muitos processos industriais em tempo real. Na literatura pode se encontrar estes tipo de sistemas em diversas indústrias, como por exemplo na indústria química, na robótica, no controlo, na indústria automóvel, em muitas mais áreas Varaiya 1993; Tomlin

et al. 1998; Perkins e Kumar 1989; Lygeros et al. 1998; Hoitomt et al. 1990; Gershwin 1989; Beydoun et al. 1998.

Um dos tipos de sistema HDS são os sistemas comutados. Este tipo de sistemas são sistemas não lineares com número finito de subsistemas ou seja é um sistema não linear dividido em diversos sistemas normalmente em vários pontos de funcionamento, com um controlador que detecta a comutação do subsistema e altera o controlador. Na literatura pode se encontrar diversos trabalhos com este tipo de sistemas, [ref livros] e Liberzon e Morse 1999.

O trabalho que foi desenvolvido nesta dissertação pode futuramente vir a ser utilizado como base de partida para controlar um sistema comutado. Com este fim sugere-se a realização de duas abordagens distintas. Assim, pode-se implementar vários controladores neuro difusos com as dinâmicas do sub sistemas de comutação e utilizar um algoritmo de detecção de comutação e comutar os controladores ou implementar um único controlador neuro difuso que se adapte à comutação. A arquitectura desenvolvida permitirá a construção das duas abordagens.

Com vista à utilização futura da primeira abordagem existe também a necessidade de investigar algoritmos de detecção do nível de comutação.

No que diz respeito ao protótipo desenvolvido em Matlab e implementado nesta dissertação, poderá futuramente ser implementado com recurso a uma interface gráfica e, por sua vez, considerar-se a realização de um estudo a nível industrial com vista à elaboração de um modelo de negócio para a sua comercialização.

Bibliografia

Abraham, A. e B. Nath (2000). "Hybrid intelligent systems design: A review of a decade of research". Em: ... *Trans-actions on Systems, Man and Cybernetics (* ... May 2014.

Berenji, H. R. e P. Khedkar (1992a). "Learning and Tuning Fuzzy Logic Controllers Through Reinforcements". Em: *IEEE Transactions on Neural Networks* 3.5, pp. 724–740.

Berenji, H. e P. Khedkar (1992b). "Learning and tuning fuzzy logic controllers through reinforcements". Em: *IEEE Transactions on Neural Networks* 3.5, pp. 724–740.

Beydoun, A., L. Y. Wang, J. Sun e S. Sivashanka (1998). "Hybrid control of automotive powertrain systems: A case study". Em: Springer, Berlin, Heidelberg, pp. 33–48.

Cox, E., M. O'Hagan, R. Taber e M. O'Hagen (1999). *The fuzzy systems handbook : a practioner's guide to building using, and maintaining fuzzy systems*. AP Professional, p. 716. ISBN: 0121944557. URL: https://dl.acm.org/citation.cfm?id=551094.

Czogala, E e J Leski (2000). *Fuzzy and Neuro-Fuzzy Intelligent Systems*. Vol. 47. Studies in Fuzziness and Soft Computing. Heidelberg: Physica-Verlag HD, p. 195.

Du, H. D. H. e N. Zhang (2008). "Application of evolving Takagi–Sugeno fuzzy model to nonlinear system identification". Em: *Applied Soft Computing* 8.1, pp. 676–686.

Emami, M. R. S. (2010). "Fuzzy Logic Applications in Chemical Processes". Em: *The Journal of Mathematics and Computer Science* 4.4, pp. 339–348.

Figueiredo, M e F Gomide (1999). "Design of fuzzy systems using neurofuzzy networks." Em: *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 10.4, pp. 815–827. DOI: 10.1109/72.774229.

Gahinet, P., A. Nemirovskii, A. Laub e M. Chilali (1994). "The LMI control toolbox". Em: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*. Vol. 3. May 1995. IEEE, pp. 2038–2041.

Gershwin, S. B. (1989). "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems". Em: *Proceedings of the IEEE* 77.1, pp. 195–209.

Ghosh, S., Q. Razouqi, H. Schumacher e A. Celmins (1998). "A survey of recent advances in fuzzy logic in telecommunications networks and new challenges". Em: *IEEE Transactions on Fuzzy Systems* 6.3, pp. 443–447.

Goh, A. (1995). "Back-propagation neural networks for modeling complex systems". Em: *Artificial Intelligence in Engineering* 9.3, pp. 143–151. ISSN: 0954-1810. DOI: 10.

1016/0954-1810(94)00011-S. URL: https://www.sciencedirect.com/science/article/pii/095418109400011S.

Hadjili, M. L. e V. Wertz (2002). "Takagi-Sugeno fuzzy modeling incorporating input variables selection". Em: *IEEE Transactions on Fuzzy Systems* 10.6, pp. 728–742.

HECHT-NIELSEN, R. (1992). "Theory of the Backpropagation Neural Network". Em: *Neural Networks for Perception*, pp. 65–93. DOI: 10.1016/B978-0-12-741252-8.50010-8. URL: https://www.sciencedirect.com/science/article/pii/B9780127412528500108.

Hirota, K. e M. Sugeno (1995). *Industrial Applications of Fuzzy Technology in the World*. Vol. 2. Advances in Fuzzy Systems — Applications and Theory. WORLD SCIENTIFIC.

Hoitomt, D. J., P. B. Luh, E. Max e K. R. Pattipati (1990). "Scheduling Jobs with Simple Precedence Constraints on Parallel Machines". Em: *IEEE Control Systems Magazine* 10.2, pp. 34–40.

Hunt, K., D. Sbarbaro, R. Żbikowski e P. Gawthrop (1992). "Neural networks for control systems—A survey". Em: *Automatica* 28.6, pp. 1083–1112. ISSN: 0005-1098. DOI: 10.1016/0005-1098(92)90053-I. URL: https://www.sciencedirect.com/science/article/pii/000510989290053I.

Inês Marques de Lucena, C. *Aplicação de técnicas de controlo óptimo difuso em ambientes distribuídos*. Rel. téc. URL: https://run.unl.pt/bitstream/10362/7702/1/Lucena{_}2012.pdf.

"Interior Point Polynomial Time Methods in Convex Programming" (2004). Em: *Spring* 42.16, pp. 3215–3224.

Jang, J.-S. (1993). "ANFIS: adaptive-network-based fuzzy inference system". Em: *IEEE Transactions on Systems, Man, and Cybernetics* 23.3, pp. 665–685.

Juang, C. F. e C. T. Lin (1998). "An on-line self-constructing neural fuzzy inference network and its applications". Em: *IEEE Transactions on Fuzzy Systems* 6.1, pp. 12–32.

Jury, E. (1996). "Remembering Four Stability Theory Pioneers of the Nineteenth Century". Em: *IEEE Transactions on Automatic Control* 41.9, p. 1242.

Kalman, R. E. (1963). "LYAPUNOV FUNCTIONS FOR THE PROBLEM OF LMI'S IN AUTOMATIC CONTROL." Em: *Proceedings of the National Academy of Sciences of the United States of America* 49.2, pp. 201–5.

Karaboga, D. e E. Kaya (2018). "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey". Em: *Artificial Intelligence Review*, pp. 1–31. ISSN: 0269-2821. DOI: 10.1007/s10462-017-9610-2. URL: http://link.springer.com/10.1007/s10462-017-9610-2.

Kasabov, N e Q Song (1999). "Dynamic Evolving Fuzzy Neural Networks with m-out-of-n Activation Nodes for On-line Adaptive Systems". Em: *The Information Science Discussion Paper Series* 99.

Kasabov, N. K. (1998). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, p. 581.

Kawamoto, S, K. Tada, A. Ishigame e T Taniguchi (1992). "An approach to stability analysis of second order fuzzy systems". Em: *Fuzzy Systems, 1992., IEEE International Conference on*, pp. 1427–1434.

Keller, M., M. Rosenberg, M. Brettel e N. Friederichsen (2014). "How Virtualization, Decentrazliation and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective". Em: *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering* 8.1, pp. 37–44.

Kriesel, D. (2005). "A Brief Introduction to Neural Networks". Em: *Retrieved August*, p. 244.

LaSalle, J. P. e S. Lefschetz (1961). *Stability by Liapunov's direct method with applications*. Elsevier Science, p. 134. ISBN: 9780124370562.

Lee, C. (1990a). "Fuzzy logic in control systems: fuzzy logic controller. I". Em: *IEEE Transactions on Systems, Man, and Cybernetics* 20.2, pp. 404–418.

— (1990b). "Fuzzy logic in control systems: fuzzy logic controller. II". Em: *IEEE Transactions on Systems, Man, and Cybernetics* 20.2, pp. 419–435.

Liapunov, A. M. A. M. e A. T. A. T. Fuller (1992). *The general problem of the stability of motion*. Tayor & Francis, p. 270. ISBN: 0748400621.

Liberzon, D. e A. Morse (1999). "Basic problems in stability and design of switched systems". Em: *IEEE Control Systems Magazine* 19.5, pp. 59–70.

Lin, C.-T. e C. Lee (1991a). "Neural-network-based fuzzy logic control and decision system". Em: *IEEE Transactions on Computers* 40.12, pp. 1320–1336.

Lin, C. T. e G. C. S. Lee (1991b). "Neural-Network-Based Fuzzy Logic Control and Decision System". Em: *IEEE Transactions on Computers* 40.12, pp. 1320–1336.

Lygeros, J., D. N. Godbole e S. Sastry (1998). "Verified hybrid controllers for automated vehicles". Em: *IEEE Transactions on Automatic Control* 43.4, pp. 522–539.

Miguel, T. e B. Oliveira (2013). "Recursive Neuro Fuzzy Techniques for Online Identification and Control". Em:

Nauck, D., F. Klawonn e R. Kruse (1997). *Foundations of neuro-fuzzy systems*. John Wiley, p. 305.

Ohr, J. (2003). *Signals and Systems ANTI-WINDUP AND CONTROL OF SYSTEMS WITH MULTIPLE INPUT SATURATIONS Tools , Solutions and Case Studies*. ISBN: 9150616919. URL: http://www.signal.uu.se/Publications/pdf/a032.pdf.

Perkins, J. e P. R. Kumar (1989). "Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing/Assembly/Disassembly Systems". Em: *IEEE Transactions on Automatic Control* 34.2, pp. 139–148.

Popov, V. M. (1961). "On absolute stability of non-linear automatic control systems". Em: *Avtomat. i Telemekh* 22.8, pp. 961–979.

Sáez, D. e R. Zúñiga (2005). "Takagi-Sugeno fuzzy model structure selection based on new sensitivity analysis". Em: *IEEE International Conference on Fuzzy Systems*. IEEE, pp. 501–506.

Scharf, E., R. Tanscheit, Q. M. C. U. of London). Department of Electrical e E. Engineering (1987). *The Application of Fuzzy Logic to Robot Control*. Research report (QMC). Department of Electrical e Electronic Engineering, Queen Mary College.

Slotine, J.-J. E., W. Li et al. (1991). *Applied nonlinear control*. Vol. 199. 1. Prentice hall Englewood Cliffs, NJ.

Sugeno, M e G. T. Kang (1988). "Structure identification of fuzzy model". Em: *Fuzzy Sets and Systems* 28.1, pp. 15–33.

Sugeno, M. e G. Kang (1986). "Fuzzy modelling and control of multilayer incinerator". Em: *Fuzzy Sets and Systems* 18.3, pp. 329–345. ISSN: 0165-0114. DOI: 10.1016/0165-0114(86)90010-2. URL: https://www.sciencedirect.com/science/article/pii/0165011486900102.

Sulzberger, S., N. Tschichold-Gurman e S. Vestli (1993). "FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks". Em: *IEEE International Conference on Neural Networks*, pp. 312–316.

Takagi, T. e M. Sugeno (1983). "Derivation of Fuzzy Control Rules from Human Operator's Control Actions". Em: *IFAC Proceedings Volumes* 16.13, pp. 55–60.

Takagi, T. e M. Sugeno (1985). "Fuzzy identification of systems and its applications to modeling and control". Em: *Systems, Man and Cybernetics, IEEE Transactions on* SMC-15.1, pp. 116–132.

Tanaka, K. e M. Sano (1993). "Fuzzy stability criterion of a class of nonlinear systems". Em: *Information Sciences* 71.1-2, pp. 3–26.

Tanaka, K. e M. Sugeno (1992). "Stability analysis and design of fuzzy control systems". Em: *Fuzzy Sets and Systems* 45.2, pp. 135–156.

Tanaka, K. e H. O. Wang (2001). *Design and Analysis Fuzzy Control Systems Design and Analysis*.

Tano, S., T. Oyama e T. Arnould (1996). "Deep combination of fuzzy inference and neural network in fuzzy inference software — FINEST". Em: *Fuzzy Sets and Systems* 82, pp. 151–160.

Teodorescu, H.-N., A. Kandel e L. C. Jain (1999). *Fuzzy and neuro-fuzzy systems in medicine*. CRC Press, p. 394.

Tomlin, C., G. J. Pappas e S. Sastry (1998). "Conflict resolution for Air Traffic Management: A study in multiagent hybrid systems". Em: *IEEE Transactions on Automatic Control* 43.4, pp. 509–521.

Varaiya, P. (1993). "Smart Cars on Smart Roads: Problems of Control". Em: *IEEE Transactions on Automatic Control* 38.2, pp. 195–207.

Willems, J. (1971). "Least squares stationary optimal control and the algebraic Riccati equation". Em: *IEEE Transactions on Automatic Control* 16.6, pp. 621–634.

Xiong, C., Y. Huang, Y. Xiong e H. Liu, eds. (2008). *Intelligent Robotics and Applications*. Vol. 5314. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg.

Zadeh, L. (1965). "Fuzzy sets". Em: *Information and Control* 8.3, pp. 338–353.